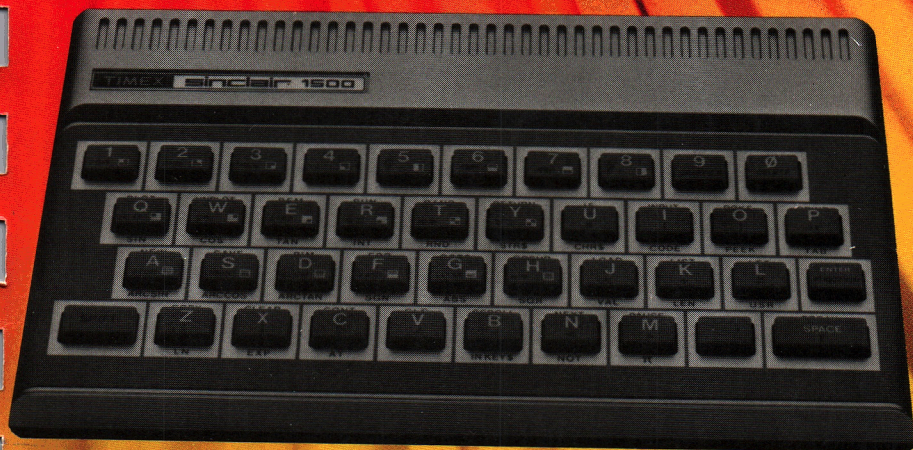
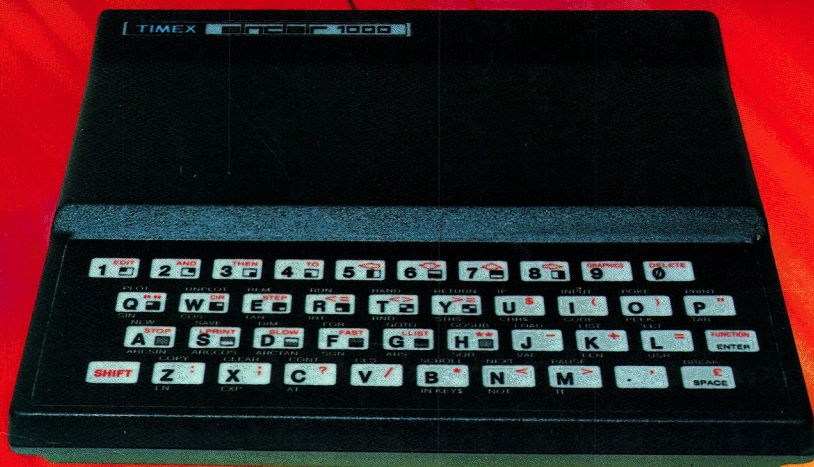


BEEPERS

21

Electronic Projects for the

TIMEX/SINCLAIR 1000 and 1500



Gordon Rockmaker and Stephen Adams

BEEPERS

21

Electronic
Projects
for the

TIMEX/SINCLAIR 1000 and 1500

BEEPERS

21

**Electronic
Projects
for the
TIMEX/SINCLAIR 1000 and 1500**

Gordon Rockmaker and Stephen Adams

McGRAW-HILL BOOK COMPANY
New York St. Louis San Francisco Bogotá
Guatemala Hamburg Lisbon Madrid Mexico Montreal
Panama Paris San Juan São Paulo Tokyo Toronto

The authors of the programs provided with this book have carefully reviewed them to ensure their performance in accordance with the specifications described in the book. Neither the authors nor McGraw-Hill, however, makes any warranties whatever concerning the programs. They assume no responsibility or liability of any kind for errors in the programs or for the consequences of any such errors.

As used in this book, the terms "Timex/Sinclair 1000 and 1500" and "T/S 1000 and 1500" refer to computers manufactured and sold by the Timex Computer Corporation. Timex/Sinclair 1000 and Timex/Sinclair 1500 are registered trademarks of the Timex Corporation. Timex/Sinclair 2000 and Timex/Sinclair 2068 are also registered trademarks of the Timex Corporation.

BEEPERS: 21 Electronic Projects for the Timex/Sinclair 1000 and 1500

Copyright © 1984 by McGraw-Hill, Inc. All rights reserved. Printed in the United States of America. Except as permitted under the Copyright Act of 1976, no part of this publication may be reproduced or distributed in any form or by any means, or stored in a data base or retrieval system, without the prior written permission of the publisher.

A BYTE Book.

1 2 3 4 5 6 7 8 9 10 S E M S E M 8 7 6 5 4

ISBN 0-07-053358-X

Library of Congress Cataloging in Publication Data

Rockmaker, Gordon.

Beepers: 21 electronic projects for the Timex/Sinclair 1000 and 1500.

1. Timex Sinclair (Computer)—Amateurs' manual.
I. Adams, Stephen, 1951- II. Title.
TK9969.R63 1984 621.3819'58 84-9639
ISBN 0-07-053358-X

The editor for this book was Jeff McCartney;
the editing supervisor was Margery Luhrs.

Contents

Preliminaries	ix
1. Getting In and Out of the Computer	1
2. Building a DC Power Supply	6
3. An Input/Output Monitor	13
4. A Logic Demonstrator	16
5. Tape Recorder Control	21
6. Beeper	25
7. A Numeric Keypad	30
8. Giant Seven-Segment Display	35
9. Scoreboard	41
10. Wheel of Fortune	44
11. Analog-to-Digital Converter	51
12. Light Pen	57
13. A Shift Lock for Keyboards	61
14. An LED Thermometer	65
15. Single-Key Operations	
Keyboard	69
16. The Moveable "Occupant"	73
17. A Computerized Voltmeter	78
18. Unbeatable Burglar Alarm	82
19. Battery Power Supply	86
20. Power Supply Filter	89
21. A Logic Probe	92

Preface

The T/S 1000 and T/S 1500 computers are powerful tools. Connected to a video screen (TV) and cassette recorder they are capable of performing the innumerable tasks one usually expects of a computer. With the hundreds of commercial software packages, print programs, and unlimited individual programs, they serve as game machines, learning instruments, teaching aids, financial managers, diagnostic tools, analysts, and data files, to name just a few functions.

But these computers can be much more, and the purpose of this book is to introduce you to some of these possibilities. It looks at the computers as intelligent engines driving a number of external devices. The 21 projects described in these pages represent only the tip of the iceberg and, in some cases, will merely whet your appetite to go further and expand the application described. Good. That's another major objective of this book: to get you to appreciate the power inside these engines and use that power to its limits.

A little knowledge of electricity and electronics is needed before you can tackle most of these projects. You will also need some of the basic tools of the electronics tinkerer. The investment is quite modest and the computers themselves are excellent values. No expensive or sophisticated instruments are required.

One word of caution: Since some of these projects involve opening the computer, and other projects involve interfacing devices not sanctioned by the manufacturer, making and connecting many of the projects in this book may invalidate the warranty on your computer.

Good luck.

Gordon Rockmaker

Preliminaries

Soldering

Many of the projects in this book require you to make a solder connection between two or more components. If you start practicing good soldering techniques now you will make fewer mistakes when you get to the projects themselves.

Caution: *Never* have the power connected to anything you are soldering; even if it does not hurt you, it will definitely damage the equipment. This applies to line voltage as well as low voltage power supplies.

Use a low wattage soldering iron (15 W) with a small soldering bit that is less likely to overheat and ruin the components. Always keep the tip of the iron clean by wiping it after using. If the tip becomes pitted, file it smooth and tin the tip again.

Tinning the iron consists of covering the tip with a thin coat of solder. The tinned tip helps transfer heat from the iron to the joint being soldered. To tin a tip, first make sure the tip is clean. Rubbing the hot tip quickly with a rag or damp sponge will help clean the tip. If the tip is pitted it should be filed smooth. Certain composition tips cannot be filed and should be replaced when corroded. Melt a few drops of solder on the clean tip. About $\frac{1}{4}$ to $\frac{3}{8}$ inch of the tip should be tinned. Do not wipe off excess solder. Allow the solder to remain on the tip for a minute or so. Then wipe off the excess solder with a clean rag. Recoat the tip with solder and immediately wipe off the excess by rubbing the tip in a

damp sponge. The tip should now be covered with a thin bright film of solder.

Keep the hot iron in some sort of metal holder and switch it off when you are not using it, even for a few minutes. This saves wear on the iron and possibly burns to both clothing and yourself.

The solder should be of the resin-core type. The thinner the gauge the better. Fine wire solder melts easier and can be used in tight locations without accidentally bridging adjacent tracks of copper or pins. Always have a container handy where you can drop surplus solder; a small aluminum can will do.

If not already tinned, all surfaces to be soldered should be tinned. Enameled insulation and oxide coating can be removed with very fine sandpaper. Commercial hook-up wire is usually tinned at the factory.

Apply the soldering iron to *both* surfaces of the joint at once. Hold the solder on the heated joint so that the joint melts the solder. The solder should flow immediately over the joint and leave it bright when cooled. Grainy, dull, grey joints are called dry or cold solder joints and make poor electrical connections. If you get a cold joint, clean it, remove excess solder and solder it again.

Remove the iron and the solder together as soon as the joint has been covered. Use a heat sink to protect the components and sockets, and prevent them from overheating and being damaged. A pair of locking metal tweezers can be used as a heat sink between the joint and the component.

After soldering the joint, cut off the surplus wire as close to the board as possible, a pair of diagonal cutters is useful, but not essential.

Practice soldering on scrap wire and defective components. Old and discarded radios and other electronic equipment are excellent for practicing soldering and unsoldering.

Schematic Symbols

The wiring diagrams used in this book show the components and how they are connected. In order to understand what each component's symbol is, the diagrams on the following page give the circuit symbol and its name.

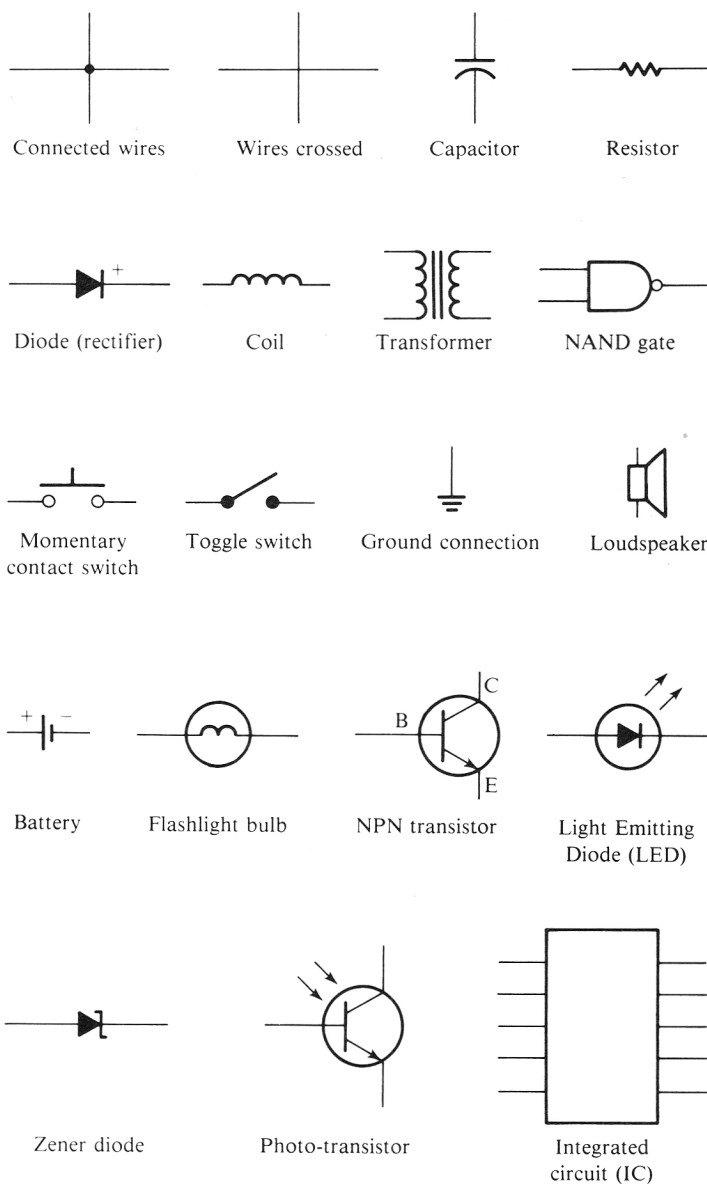


Fig. P-1



PROJECT 1

Getting In and Out of the Computer

In order to make the T/S 1000 and T/S 1500 more useful, and allow them to control external devices, we must first be able to send signals to and from these computers. A device to do this is called an **interface**.

An INPUT/OUTPUT Port

A common interface is an INPUT/OUTPUT port. This consists of one or several integrated circuits which will store any data sent to it and keep it available for an external device. The maximum amount of data that it can store is eight binary digits (called bits). The information is available at eight points, or taps, each of them having either +5 or 0 volts (V) on it. The INPUT and OUTPUT ports are usually separate, so the data emitted by the OUTPUT port is not affected by “reading” the INPUT port. To tell if it is a READing (INPUT) operation or a WRITEing (OUTPUT) operation the computers put out two signals NOT WRITE ($\overline{WR}$) and NOT READ ($\overline{RD}$). The fact that either of these signals is at 0 V enables the operation to be done.

The external device also requires a place where data can be collected and sent. The place is called an *ADDRESS*. The ADDRESS applies only to the INPUT/OUTPUT port and no other piece of equipment connected to the computer. When the computer wants to talk to the port, it sends the number representing the ADDRESS over 16 ADDRESS wires, labeled

A0 to A15. Since the T/S 1000 and T/S 1500 are not supplied with a user's port, one must be externally attached.

One type of port is called a *memory mapped* port. It is treated like a piece of RAM (Random Access Memory). That is, it may be **PEEK**'d (transferred from the port into the program) or **POKE**'d (transferred to the port from the program).

The **PEEK**, **POKE**, and **BASIC** commands can be inserted in a program in the following form:

PEEK 16396

POKE 16396,255

PEEK returns the number between 0 and 255 (the maximum number of combinations available from 8 bits) that is in the memory location indicated by the number after the word **PEEK**. **POKE** puts a number into the memory. The first number after the word **POKE** is the memory location; the number after the comma is the number (between 0 and 255) that will be placed in that memory location. No matter what method you use, you can only put in a number between 0 and 255. This is because the highest decimal number that can be represented by 8 bits is 255.

Figure 1-1 represents an 8-bit number. The binary number system is used to represent the value in each bit position. This numbering system uses only the digits 0 and 1. A one in a bit position means that the bit position has the decimal value shown in the line above the bit box. A zero in a bit position means a decimal value of zero. The total decimal value of the 8-bit number is simply the sum of all the decimal values in the individual bit positions.

Each bit location represents a power of two. The bit num-

Decimal value	2^7	2^6	2^5	2^4	2^3	2^2	2^1	2^0
Binary number	1	0	1	0	1	0	0	1
Bit position	Bit 7	Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0

Fig. 1-1

ber gives the number of times that 2 (in the decimal system) must be multiplied by itself if there is a binary 1 in that location. We will illustrate the process with the example shown in Figure 1-1. Bit 3 represents 2 multiplied by itself three times, or $2 \times 2 \times 2$ or simply 2^3 (two to the third power or two cubed). Since a binary 1 occupies that position, the 1 represents $2^3 = 8$ in the decimal system (that's the system we use in our everyday arithmetic). To find out what the entire 8-bit binary number represents in our decimal system, we need to add the values of each of the bit locations.

$$\begin{aligned}
 \text{Bit } 7 &= 2^7 \text{ (contains binary 1)} = 128 \\
 \text{Bit } 6 &= 2^6 \text{ (contains binary 0)} = 0 \\
 \text{Bit } 5 &= 2^5 \text{ (contains binary 1)} = 32 \\
 \text{Bit } 4 &= 2^4 \text{ (contains binary 0)} = 0 \\
 \text{Bit } 3 &= 2^3 \text{ (contains binary 1)} = 8 \\
 \text{Bit } 2 &= 2^2 \text{ (contains binary 0)} = 0 \\
 \text{Bit } 1 &= 2^1 \text{ (contains binary 0)} = 0 \\
 \text{Bit } 0 &= 2^0 \text{ (contains binary 1)} = 1 \\
 10101001 \text{ (binary)} &= 169 \text{ (decimal)}
 \end{aligned}$$

That last one, bit 0, is a little tricky. It means 2 is multiplied by itself zero times. In math that's defined as 1. In fact, any number multiplied by itself zero times (that is, raised to the zero power) is defined as 1. For purposes of converting from the binary to the decimal number system all we need remember is that $\text{bit } 0 = 2^0 = 1$ (*not* $2 \times 0 = 0$).

Converting a decimal number to an 8-bit binary is a simple process also. We will convert 169 (decimal) to its equivalent 8-bit binary number. The procedure is as follows.

Divide the number and each of the succeeding quotients by 2. The remainder in each case is the binary number that goes into each bit location starting with bit 0.

$$\begin{aligned}
 169 \div 2 &= 84 \text{ R}1 & 1 \text{ (bit 0)} \\
 84 \div 2 &= 42 \text{ R}0 & 0 \text{ (bit 1)} \\
 42 \div 2 &= 21 \text{ R}0 & 0 \text{ (bit 2)} \\
 21 \div 2 &= 10 \text{ R}1 & 1 \text{ (bit 3)} \\
 10 \div 2 &= 5 \text{ R}0 & 0 \text{ (bit 4)} \\
 5 \div 2 &= 2 \text{ R}1 & 1 \text{ (bit 5)}
 \end{aligned}$$

4 / Project 1

$$2 \div 2 = 1 \text{ R}0 \quad 0 \text{ (bit 6)}$$

$$1 \div 2 = 0 \text{ R}1 \quad 1 \text{ (bit 7)}$$

When the quotient is 0 the process stops. Note that the answer 10101001 agrees with our previous example.

Here's another example. Convert 28 (decimal) to an 8-bit binary number.

$$28 \div 2 = 14 \text{ R}0 \quad 0 \text{ (bit 0)}$$

$$14 \div 2 = 7 \text{ R}0 \quad 0 \text{ (bit 1)}$$

$$7 \div 2 = 3 \text{ R}1 \quad 1 \text{ (bit 2)}$$

$$3 \div 2 = 1 \text{ R}1 \quad 1 \text{ (bit 3)}$$

$$1 \div 2 = 0 \text{ R}1 \quad 1 \text{ (bit 4)}$$

At this point the process stops so that bits 5 through 7 are zero.

$$28 \text{ (decimal)} = 00011100 \text{ (binary)}$$

At the back of the T/S 1000 and T/S 1500 is a 23-point edge connector which connects to each of the computer's Z80 microprocessor lines. The edge connector is exposed at the left side when looking at the back of the T/S 1000. In the T/S 1500 it is in about the same position except that a small

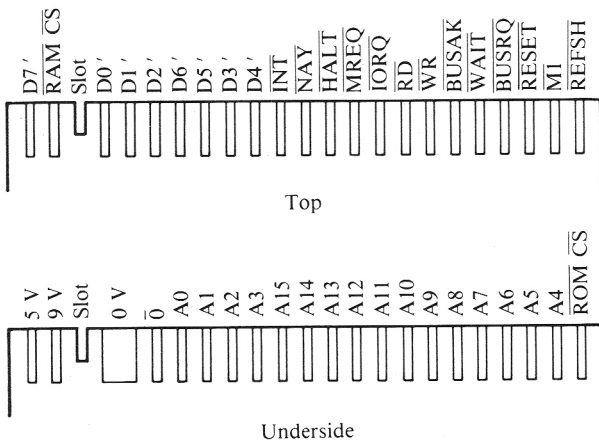


Fig. 1-2

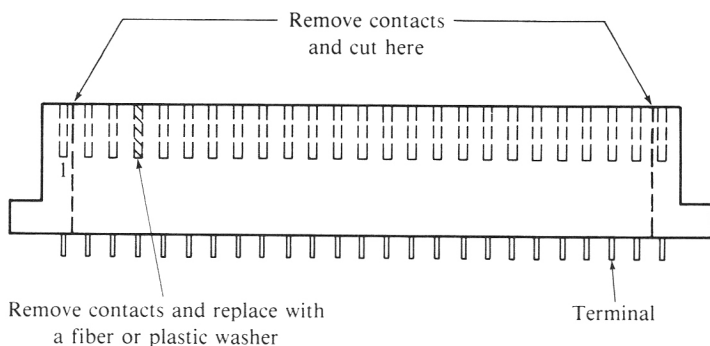


Fig. 1-3

hinged lid covers the area. Looking down at the connector, each of the lines to the microprocessor are as shown in Figure 1-2.

In order to access this edge connector directly for any of the projects or to build extension cables to any of the interfaces (for example, a printer or RAM expansion module) it will be necessary to make a socket to fit the edge connector. While 23-pin connectors can be purchased from suppliers specializing in Timex/Sinclair parts and components, these are usually made from standard 25-pin connectors.

An all-plastic 25-pin female edgeboard connector is shown in Figure 1-3. Pins 1 and 25 can be cut off using a medium-coarse hacksaw blade. Once the connector is cut, the two pins in new position 3 can be pulled out using a long-nose plier. A fiber or plastic wedge can then be forced into the space to serve as the key to fit into the computer's edge connector slot. A thin washer such as those used in faucets will do the job well. A standard 50-conductor ribbon cable of the required length can then be soldered or wire-wrapped to the connector's terminals.

PROJECT 2

Building a DC Power Supply

To use most of the projects in this book an extra source of power is required. The +5-V internal supply of the computer is designed to handle only the computer power requirements. This project, therefore, is to build a power supply that will provide +5 V dc to run the integrated circuits and +12 V to operate lamps, relay, and the like.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Transformer, 120 V/12 V, 2 A
1	Bridge rectifier, 6 A
1	Electrolytic capacitor, 2000 μ F, 35 V
1	IC 7805, voltage regulator
1	IC 7812, voltage regulator
2	Capacitor, 0.1 μ F, 35 V
	Project chassis or case, screws, and nuts as required

Construction

The design of the power supply is based on two voltage regulator ICs, the 7805 and the 7812 (Figure 2-1). These devices handle considerable energy for their size. Therefore, a means must be provided to dissipate any excess energy. The excess energy is in the form of heat energy and a device called a *heat sink* is used to draw the heat away from the regulator and

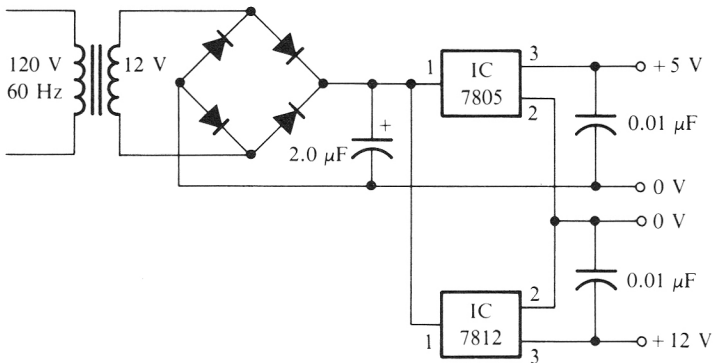


Fig. 2-1

dissipate it into the surrounding air. Basically the heat sink is a piece of metal with good heat-conducting properties and a large surface for radiating the heat. In this project the heat sink will be a sheet of 18 gauge aluminum, 4 inches square. Figure 2-2 shows how the heat sink is constructed and how the regulators are attached to it.

The metal tab that conducts the heat to the heat sink is also the ground connection for the regulator. Therefore, care

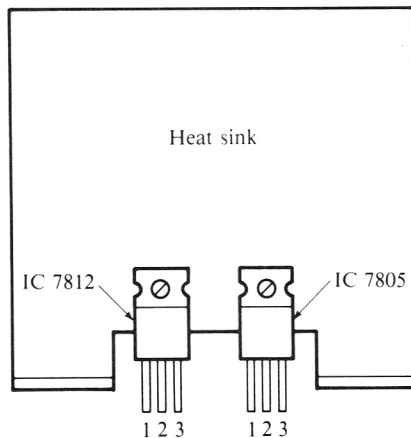


Fig. 2-2

must be taken not to have other wires touch the heat sink unless they are also to be grounded. The heat sink should be mounted in the power supply housing or on a board so that it is well ventilated and can readily dissipate the heat generated by the regulators. In addition, the heat sink should be insulated from the rest of the power supply circuit. This can be done by first mounting the heat sink on a nonmetallic base.

A step-down transformer converts the voltage at your wall outlet (usually about 120 V) to 12 V. Since it does carry the line voltage, it should be handled with extreme care. The wires that connect to the line are attached to the *primary* side of the transformer. Some transformers may be purchased with a line cord connected to their primary. The transformer is then simply plugged into the wall much like any other electrical appliance. No special precautions about its use are necessary. The transformer wires that connect to the power supply circuit are attached to the transformer's *secondary* winding.

In this project the secondary voltage is 12 V. While any electrical voltage has the potential to do damage, low voltages are rarely a threat to life. However, if not handled properly, they can be short circuited creating the dangers of sparks and fire.

Caution: Before wiring the power supply, make sure the transformer is not plugged into an outlet. All power must be off when handling the power supply circuit.

If the transformer does not have its own line cord, one can be purchased and soldered to the two primary wires. Line cords are often available with a rubber or plastic molded plug on one end and two bare tinned wires on the other end. The two tinned wires can be soldered to the transformer's two primary wires. Usually the transformer wires will already have been tinned. If not, tin them before soldering to the line cord.

A convenient way to insulate the solder joint is to use heat-shrinkable tubing or spaghetti. A small length of tubing about $\frac{1}{2}$ inch longer than the splice itself is slipped on the wire before the splice is made. After the splice is soldered, the tubing is slipped over the splice, overlapping on the wire's insulation on both ends. Heat is then applied to the tubing which will shrink tightly around the splice and wires.

The rest of the construction of the power supply is a matter of choice, as long as the schematic diagram is followed. Care must be taken to see that the polarity markings on the components are followed. It is particularly critical to follow polarity markings of electrolytic capacitors.

How It Works

The usual voltage at the receptacles in the home and at the office is about 120 V, although in any particular place it may range between 110 and 125 V. The type of voltage is called alternating current or simply ac. Don't be concerned about a voltage being called a current, it's part of the jargon of the trade and completely acceptable. The main point is that, whether it's a voltage or a current, it is alternating or varying with time.

If we measured the voltage in an ac circuit every fraction of a second for an entire second and plotted the values on a graph, the graph would look something like Figure 2-3. The vertical line is a scale of voltage values and the horizontal line is a scale of time. A voltage of this sort can't be used directly on our electronic component for two reasons: its value is too high (all of the parts would probably burn out), and its value is not constant. The purpose of the power supply is to reduce the value of the line voltage and to smooth out the varying voltage to an even, steady value.

The transformer will step down the line voltage from 120 to 12 V ac. Transformers work only on ac. They consist of coils of wire wound around iron cores. If the transformer consists of two coils, as in our case, the number of turns of wire of one coil divided by the number of turns of wire of the other coil gives us the *turns ratio*. This is also the *voltage ratio* of the transformer, that is, the ratio of one voltage to another. In our case, the input voltage (called the primary) is 120 V and the output voltage (called the secondary) is 12 V. The voltage ratio is therefore

$$\frac{120 \text{ V}}{12 \text{ V}} = 10$$

This also tells us that there are ten times more turns of wire in the primary than in the secondary. But we still can't use

the 12 V since it is ac (that is, it's varying in value with time). The next job is to keep the polarity from changing with time. We do this with a bridge rectifier.

A rectifier is made of a material that allows current to flow through it in only one direction. In an ac circuit, the polarity changes every half cycle. With a rectifier in the circuit, current will flow only when the polarity of the circuit matches the polarity marked at the terminals of the rectifier. As a result, the rectified waveform looks like that of Figure 2-4. The trouble with using a single rectifier is that current only flows during every other half cycle. To solve this problem, four rectifiers are connected in what is known as a bridge circuit. When properly connected across the ac line, each half cycle of the ac wave is rectified as in Figure 2-5. While four individual rectifiers can be hooked up as shown, bridge rectifiers are available with all four devices connected internally and only the four circuit connections brought out.

The output of the bridge rectifier is a pulsating dc. That is, the polarity is constant, but the value varies continuously from zero to a maximum value and back to zero. What is now needed is a method of smoothing out this pulsating wave so that the voltage has a constant value of about 12 V. The smoothing process is accomplished by connecting an electrolytic capacitor across the output of the bridge rectifier. As the voltage from the rectifier builds from zero to its maximum value, the capacitor stores energy. We say the capacitor is being charged. When the rectifier output starts dropping toward zero, the capacitor returns some of its energy to the circuit. As a result, the voltage of the circuit drops only slightly. Before the capacitor can discharge its energy to a low level, the rectifier output is building up to its maximum again. This restores the capacitor to its fully charged state and the entire process of charge and discharge is repeated. Figure 2-6 shows how this process works. The solid line represents the actual voltage across the line. Notice that the dc still varies slightly. This is called ripple. The voltage regulators eliminate almost all of the ripple and maintain a fairly accurate output voltage: 5 and 12 V in our case. The ceramic capacitors at the output of the power supply serve to protect the voltage regulators

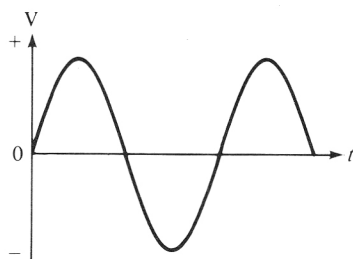


Fig. 2-3

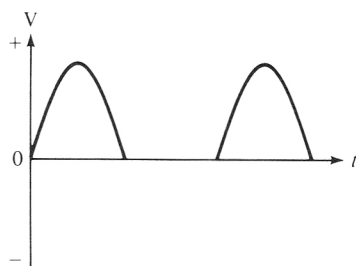


Fig. 2-4

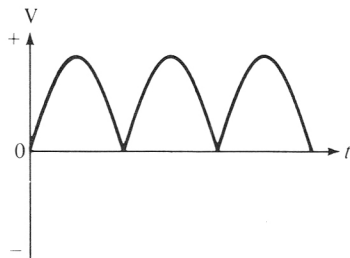


Fig. 2-5

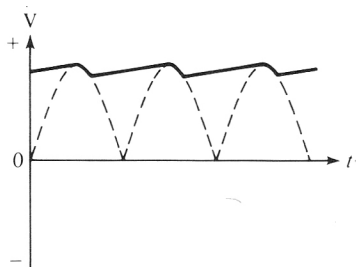


Fig. 2-6

from any high voltage surges that may originate from the load side of the circuit.

The voltage regulators used in this supply can each handle 1 amp (A). There are regulators that are rated at higher currents. Remember, however, that all the components of a circuit must be capable of handling the same load. If regulators having a higher current rating are desired, the transformer and bridge rectifier must be sized to take this higher current.

PROJECT 3

An Input/Output Monitor

There are 16 lines available at the INPUT/OUTPUT port: eight lines for input and eight lines for output. These lines are not connected to the computer data bus and are not involved with processing unless the T/S 1000 or T/S 1500 talks to the INPUT/OUTPUT port. To test the status of these lines a monitor can be connected to the port with an appropriate display.

In this project you will build a display consisting of a set of LEDs that will monitor the status of each of the 16 INPUT/OUTPUT lines.

Materials Required

<i>Quantity</i>	<i>Material</i>
2	IC 7400
8	Resistor, 1.2 k Ω , 1/4 W
8	LED (any color or shape)
1	DIP switch, single-pole, 8-station
2	IC socket, 14-pin
1	INPUT/OUTPUT port
1	Power supply, +5 V dc, 100 mA

How It works

Figure 3-1 is the schematic diagram for this project.

Output Monitor. A 7400 IC is used at the output section of the port. This IC is made up of four NAND gates. Each

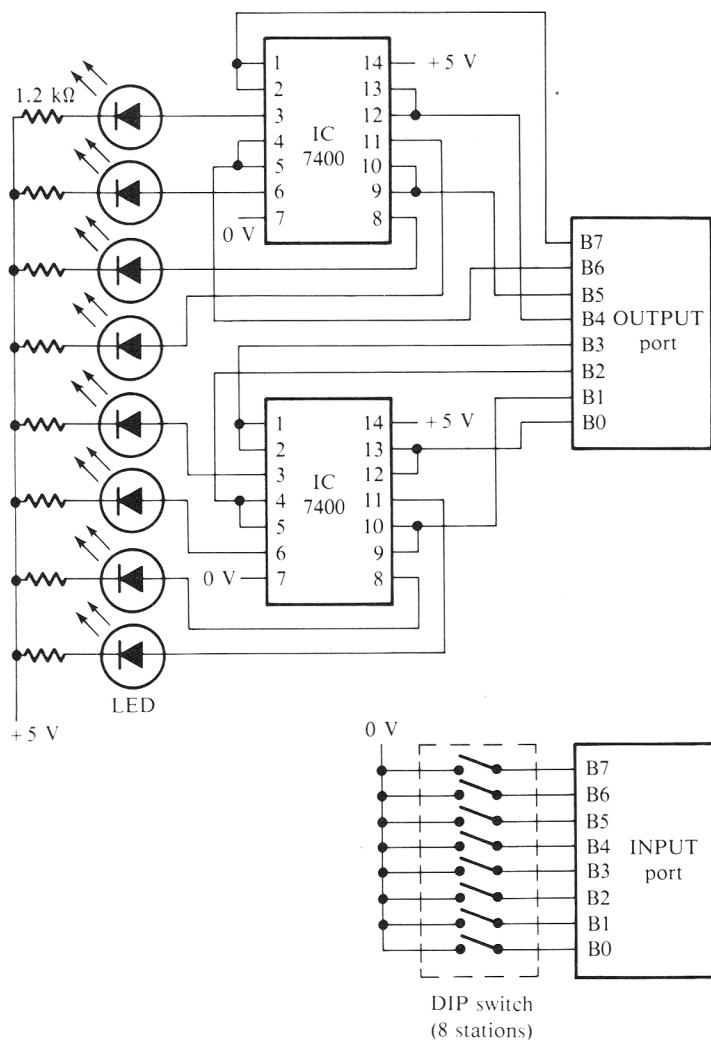


Fig. 3-1

of the outputs of the gates drives an LED through a resistor. The purpose of the resistor is to limit the current through the LED and thus prevent it from burning out. The two inputs to each of the NAND gates in the IC are connected to-

gether so that the gates function merely as inverters. An inverter changes the status of its input. Thus a binary 1 or HI at the input of the inverter will appear as a binary 0 or LO at the inverter's output, and a binary 0 at the input of the inverter would be inverted to a binary 1.

The inputs to the IC are the OUTPUT lines of the port. Four of the OUTPUT lines are the inputs to one of the ICs and the other four lines are the inputs to the second IC. When the output of the inverter is a binary 0, current will flow in the LED circuit connected to that particular inverter and the LED will turn on. In order to obtain a binary 0 at the inverter's output, the inverter's input must be a binary 1. The inverter input will get its binary 1 from the OUTPUT port line to which it is connected. Thus a lit LED indicates a binary 1 in one of the OUTPUT lines.

Input Monitor. Each of the eight lines of the INPUT port is connected internally to the +5-V supply through resistors. This produces a HI or binary 1 on each of the lines. **PEEKing** the status of these lines would produce a decimal 255 since the eight bits would be 11111111. The input monitor consists of eight single-pole single-throw switches, that is, simple ON-OFF switches, one for each INPUT line. One side of the switch is connected to the INPUT line, the other side of the switch is connected to all the other switches which in turn are connected to ground or the 0-V side of the power supply.

When a switch is closed, that particular line is brought down to 0 V or LO. This is binary 0. By again **PEEKing**, the status of the INPUT port will be returned, indicating which bit or bits are zero. For example, if upon **PEEKing**, the decimal 197 were returned, the binary number 01100101 would be indicated. Thus the switches connected to B1, B3, B4, and B7 must be closed. Try different combinations of open and closed switches and compile a list of the numbers you have **PEEK'd**. Although the switch combinations produce binary numbers, the T/S 1000 and T/S 1500 change them to the equivalent decimal numbers which are displayed on the screen. Similarly decimal numbers being sent to the OUTPUT port are first converted to binary numbers by the computer.

PROJECT 4

A Logic Demonstrator

The purpose of this project is to make a demonstration model of a logic gate. The computer can use the gate for anything you wish. There are three inputs (B0-B2) to the INPUT port, but there could be as many as eight, each one using one bit. Each INPUT bit can be set to binary 1 or binary 0 using the switch provided. Internal resistors in the INPUT-port chip mean that if nothing is connected to that bit, it is set to binary 1 (+5 V). By closing the switch connected to 0 V this input can be changed to binary 0. The switch position is marked on the board as an input to the gate (which is drawn on a piece of paper).

The computer, having computed what the output should be, lights the LED if the output is binary 1 (Figure 4-1). The logic of the gate can be changed by calling a different sub-routine within the given program, and changing the piece of paper with the gate drawn on it. The gate could be selected by a random number and a pupil asked to identify the gate by analyzing the results of changing the switch combinations.

This model can be expanded up to eight inputs and up to eight outputs. As a result of using the computer to do the logic of the gate, quite a complex network can exist between the inputs and the outputs. If you intend to use more than one LED as an output, then it is best that you do not attach them directly to the OUTPUT port or it may cause damage. You can safely drive up to four LEDs from a 7400 IC as shown in the Monitor project (Project 3). The 7400 and

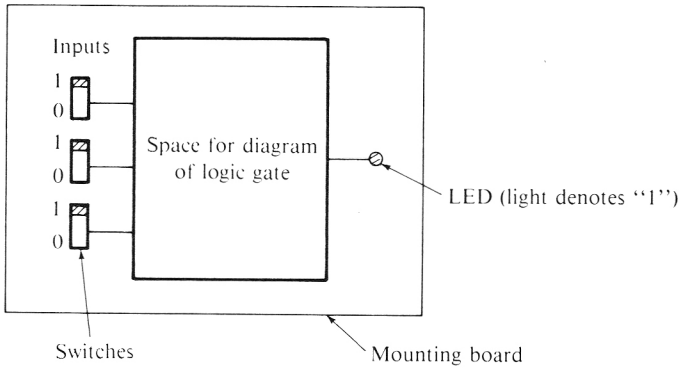


Fig. 4-1

74LS00 are exactly the same except that the 74LS00 draws less power to operate the chip.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	INPUT/OUTPUT port
1	LED
1	Resistor, 220 Ω , 1/4 W
3	Toggle switch, single-pole double-throw
	Mounting board, screws and nuts as required

Construction

In Figure 4-2 a 220- Ω resistor limits the current flowing through the LED and the OUTPUT port. It is required to stop any damage to the port or the LED by excessive current from the +5-V line, through the OUTPUT chip to the 0-V line via the resistor and LED. The LED can be any color or shape. Its cathode, which is connected to the 0-V line, can be identified by the fact that it is larger, and nearer the flattened portion of the plastic encapsulation, than the anode. All diodes have one cathode and one anode.

The circuit uses single-pole double-throw toggle switches.

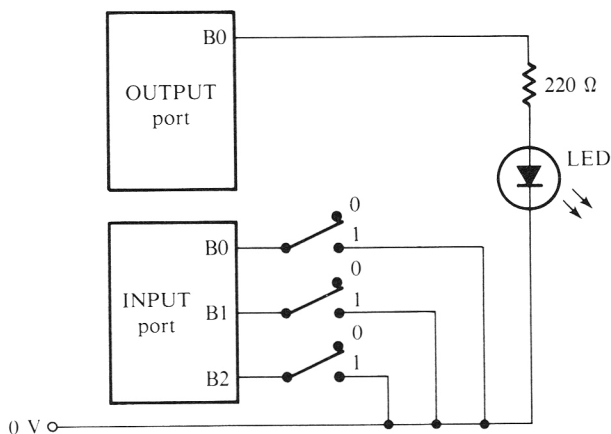


Fig. 4-2

The wiring to the computer INPUT port should be kept short (not more than 2 feet), as longer wires would pick up extraneous signals from radios and other equipment. These signals could cause a wrong reading of the switches if they were left in the binary 1 position. As long as there is a current flowing (such as, when the switches are connected to 0 V), this will not affect the input; but when an INPUT port is just connected to a long piece of wire, it can act like a radio antenna picking up all sorts of electromagnetic interference.

The Program

```

10 DIM N(8)
20 LET IN=PEEK PORT
30 REM USE USR IF NECESSARY
40 REM GET BITS
50 FOR X=7 TO 0 STEP -1
60 IF 2**X>IN THEN GOTO 90
70 LET N(X+1)=1
80 LET IN=IN-(2**X)

```



```

90  NEXT X
100 PRINT "2.AND 3.OR 4.NAND 5.NOR"
110 INPUT IN
120 IF IN>5 OR IN<2 THEN GOTO 110
130 GOSUB IN*100
140 POKE PORT, X
150 REM USE USR IF NECESSARY
160 INPUT A$
170 CLS
180 GOTO 10
200 LET X=N(1) AND N(2) AND N(3)
210 RETURN
300 LET X=N(1) OR N(2) OR N(3)
310 RETURN
500 REM INVERTER COMES HERE
510 GOSUB (X-2)*100
520 LET X=NOT X
530 RETURN

```

Lines 50-90 allocate a 1 to the appropriate $N(X + 1)$ where X is the bit being tested. The test consists of finding out if the number being tested is equal to, or greater than, the value in decimal form for that bit. Since **DIM** $N(8)$ sets all eight numbers to zero, any number which comes through the jump to line 90 will be a 0. If $N(X + 1) = 1$, then the equivalent decimal number for that bit is subtracted from the total before testing the next bit.

Lines 100-130 give the operator a choice of gates, and select by calculating the subroutine from the **INPUT** IN . If the input is 4 or 5, subroutine 500 is called. The T/S 1000 and T/S 1500 BASIC assumes that if it cannot find line 400, then the next line is the correct one. In this case it is line 500.

Line 140 lights the LED by setting bit 0 to binary 1 if $X = 1$. Subroutine 500 calls the appropriate routine, then inverts the result, for example, if $X = 1$ then **NOT** $X = 0$, and if $X = 0$ then **NOT** $X = 1$. The entire program requires only 1K of memory.

PROJECT 5

Tape Recorder Control

The tape recorder control shows how the T/S 1000 and T/S 1500 can control red relays and how useful they can be. A reed relay can isolate the controlling equipment from the equipment to be controlled. This allows the voltages of the two systems to be totally different. The current-carrying capacity is also limited only by the capacity of the switch. The reed we are using can carry up to 125 V, dc or ac, and a maximum current of 1 A.

What is a reed relay? It consists of a glass tube containing two fine magnetic strips, each having the same magnetic pole at its tip. As like poles repel, the strips are kept apart. The switch is, therefore, open. When another magnetic field is introduced, it overcomes the strips' magnetism and the switch closes. The glass tube contains a vacuum, which means that

Materials Required

<i>Quantity</i>	<i>Material</i>
1	OUTPUT port
1	IC 7400 or 74LS00
1	Resistor, 68 Ω , 1/4 W
1	Reed switch, normally open
70 ft	Magnet wire, No. 40
1	Jack plug, 2.5 mm
1	Diode
1	Power supply, +5 V dc

the contacts never get dirty or corrode. The external magnetic field may be supplied by a simple bar magnet or an electro-magnet; we shall use the latter one.

Construction

The schematic diagram for this project is Figure 5-1. Our magnet will consist of a coil of wire which, when dc is passed through it, produces a magnetic field. The coil is wound directly over the glass reed and consists of 1600 turns of No. 40 enamelled magnet wire (Figure 5-2). The No. 40 wire is quite thin and the enamel coating insulates the wire. The enamelling needs to be scraped off the wire at both ends of the coil (using very fine sandpaper) before you will be able to solder it. The coil is best wound using a hand drill with the handle held in a vise. The reed is first insulated from the coil by winding one layer of insulating tape around it. Then it is placed in the jaws of the drill and *lightly* gripped. Tape the end of the wire to the drill chuck and start to turn the handle. Hold the wire taut with one hand and turn the handle of the

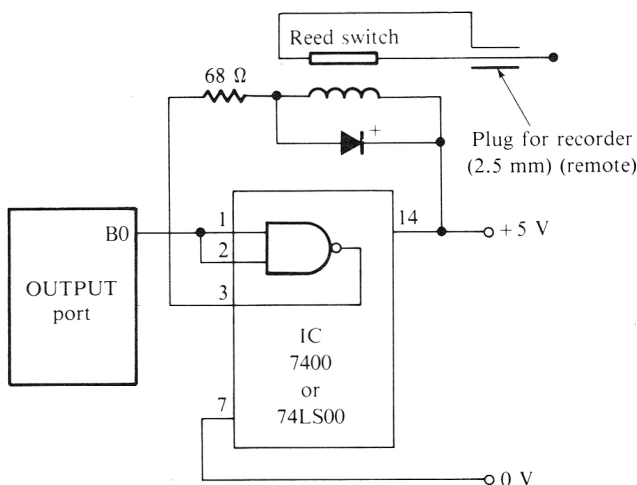
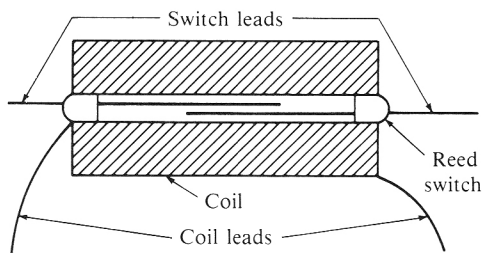


Fig. 5-1



CROSS SECTION OF REED SWITCH AND COIL

Fig. 5-2

drill with the other, remembering that one turn of the handle gives many turns of the reed. The coil does not have to be absolutely precision wound; a few turns more or less will make no difference in its operation. When the winding is complete, wrap a layer of insulating tape over the coil to protect it and prevent it from unwinding.

Once the reed switch and coil are completed, the rest of the circuit can be started. Only one gate of the IC is shown (there are four in total); the other gates can be used to drive more reed relays if required. The connections are shown in the Monitor project (Project 3). The 68- Ω resistor limits the current through the gate, but there is enough current to operate the coil. The diode is there just to protect the IC. When the current to the coil is turned off, the magnetic field collapses, and in doing so creates another voltage, opposite to the one initially created. Although this voltage only exists for a few microseconds, the voltage created can be quite high. The diode, being a one-way device, will not affect the circuit when the coil is operating, but when the reverse voltage occurs it conducts, shorting the voltage out.

The circuit works as follows: The bit 0 of the OUTPUT port goes to binary 1 which, because the gate inverts the input signal, sets the output of the gate to 0 V. As the end of the coil is now at 0 V, current flows through the coil from the +5-V line, closing the switch. The closing of the switch turns on the cassette motor ready for **LOADing** or **SAVEing**. When the bit 0 is returned to binary 0, the output of the gate is

restored to +5 V and, there being +5 V on both ends of the coil, the switch opens. This turns off the motor of the cassette recorder.

The following information shows how to make use of this in programs. It may also be used to **SAVE** and **LOAD** data if you want to write your own machine code routines.

The Program

SAVE

```
2000  PRINT "PRESS PLAY AND RECORD  
      BUTTONS"  
2010  PRINT "THEN PRESS NEW LINE"  
2020  INPUT A$  
2030  POKE PORT, 1  
2040  REM USE USR IF NECESSARY  
2050  SAVE "TAPE"  
2060  POKE PORT, 0  
2070  REM REST OF PROGRAM
```

LOAD

```
1000  POKE PORT, 1  
1010  LOAD "TAPE"  
1020  REM REST OF PROGRAM IGNORED
```

PROJECT 6

Beeper

One of the disadvantages of the T/S 1000 and T/S 1500 is not having a means of making sounds to accompany games and devices. This project provides a cheap, one-note oscillator which can be turned ON and OFF by an OUTPUT port. Only one bit is used, so the other bits can be used to control other things, even seven more one-note oscillators.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	OUTPUT port
1	Loudspeaker, 8 Ω
1	Resistor, 18 k Ω , 1/4 W
1	Resistor, 100 Ω , 1/4 W
1	Electrolytic capacitor, 1 μ F, 15 V
1	Capacitor, 0.01 μ F
1	IC NE555
1	Power supply, +5 V dc, 200 mA

Construction

The whole circuit (Figure 6-1) can be built on a small piece of perfboard, complete with loudspeaker and attached to the port by one piece of wire. The 0-V line of the power supply must, of course, be attached to the 0-V line of the T/S 1000 or T/S 1500.

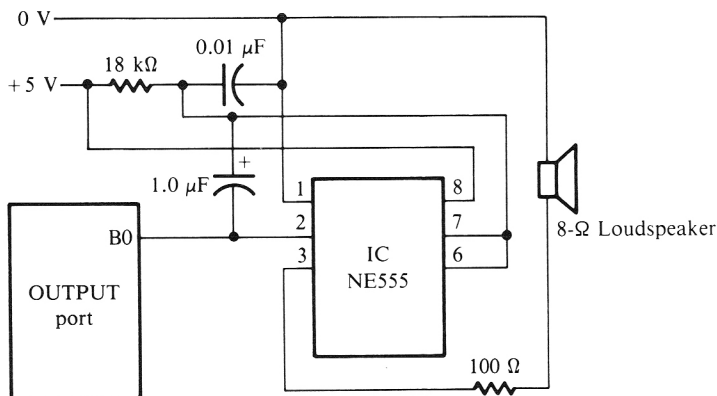


Fig. 6-1

If you wish, or the family gets upset with your *beeping*, you can change the loudspeaker to a personal earphone. This should be of 8-Ω impedance and should be of the same type as can be used for portable radios, tape recorders, and the like.

The 1-μF capacitor is not a critical component and could easily be changed for another of lower value. It need not be an electrolytic capacitor; try a substitute and see. The 18-kilohm (kΩ) resistor controls the frequency of the tone generated but a 50-kΩ resistor could be substituted, in order to alter the frequency. The IC should be fitted into a socket if you are not confident about soldering it. Although the IC is quite heat tolerant, using a socket also means that you can check all the connections before inserting the IC. The layout of the components is not critical; we have left that up to you.

How It Works

Before we have a look at what the 555 IC does, it is important to find out something about capacitors, as the circuit depends on their use.

A capacitor is like a large water tank. If the tank (or ca-

pacitor) is empty, it is said to be *discharged*. To fill the tank with water, we need to open a tap which will allow a flow of water into the tank. Similarly, electrons can be allowed to flow into the capacitor. We call this *charging* the capacitor. Just as it takes time to fill the tank, it takes time to completely charge the capacitor. In an electric circuit, the amount of charge the capacitor can take is called *capacitance*. The unit of capacitance is the *farad*, abbreviated F. The flow of current into the capacitor will stop just as soon as the capacitor is fully charged. The farad is a large unit; in fact, a capacitor able to hold a charge of 1 F would be as large as a room. Therefore, a much smaller unit of capacitance is used. The smaller unit is the *microfarad* which is one-millionth of a farad.

What we are interested in, however, is the *time* it takes to charge the capacitor. This can be found by multiplying the *capacitance* of the capacitor in *farads* by the *resistance* of the resistor in *ohms*. If the resistance is lowered or capacitance is increased (just like increasing the capacity of the tank or further opening the tap), the time it takes to charge the capacitor will decrease. In an electric circuit it is easier to change the value of resistance than to vary capacitance significantly.

The NE555 IC is called a *timer* because of the periodic cycling of its output. It compares its input (pin 6) constantly with its supply voltage (in this case +5 V) and, if it is lower than two-thirds of the supply voltage, it outputs +5 V on pin 3. When the input reaches two-thirds of the supply voltage, the IC connects pin 7 to 0 V and changes the output to 0 V. This will now stay in this position until it receives a pulse of 0 V on its pin 2. The pulse is a brief drop in voltage from its normal +5-V value down to 0 V and back up to +5 V. This is called the *trigger* input pin and it will operate in reaction to pulses as short as 1 microsecond, that is, one-millionth of a second. The IC then outputs +5 V on pin 3 again and starts the whole process over again. The time taken to reach two-thirds of the supply voltage will depend on how long the capacitor takes to charge on pin 6 and pin 7. The capacitor voltage rises as the capacitor charges. When pin 7 changes to 0 V, the capacitor will be completely discharged. At this point

there is 0 V on both sides of the capacitor. In effect the capacitor is shorted. The capacitor will not be able to charge up again until this short is removed. When pin 2 is pulsed to 0 V, the capacitor starts to charge again. The circuit only charges up the $0.01\text{-}\mu\text{F}$ capacitor, as the $1\text{-}\mu\text{F}$ capacitor is already connected to +5 V at pin 2, and is discharging (as the other end is rising to +5 V). Having the same voltage at both ends of the capacitor is just like shorting it out, remember?

As the $0.01\text{-}\mu\text{F}$ capacitor charges, the voltage on pin 6 rises; when it has reached two-thirds of the supply voltage (3.33 V), the discharge circuit operates and discharges the $0.01\text{-}\mu\text{F}$ capacitor, providing pin 2 with the pulse it needs to get started again. This is because pin 2 starts off at 0 V (initially the $1\text{-}\mu\text{F}$ capacitor is in a discharged state) and as the $1\text{-}\mu\text{F}$ capacitor charges up, pin 2 slowly returns to +5 V. All this charging and discharging of the $0.01\text{-}\mu\text{F}$ capacitor causes the output to fluctuate between +5 V and 0 V. This changing voltage causes a change in the current flowing in the loudspeaker. Thus the loudspeaker cone moves in and out, the result being the beeping tone, at a frequency determined by the capacitor's charging and discharging rate.

The frequency of this tone can be calculated by dividing one by the time (in seconds) taken to charge the capacitor to two-thirds of the supply voltage. This will give us the number of cycles in 1 second, which is the frequency. The unit of frequency is the hertz, abbreviated Hz. The frequency formula is

$$f = \frac{1}{R \times C}$$

where R is the resistance in ohms and C is the capacitance in farads.

In our example

$$f = \frac{1}{18 \times 10^3 \times 0.01 \times 10^{-6}} = 5555 \text{ Hz}$$

The next question is: How do we control the *beeping* from the computer? If pin 2 is at +5 V (binary 1), then all works

normally, and this is guaranteed by connecting it to the OUTPUT port B0. If, however, bit 0 of the port changes to 0 V (binary 0), then pin 2 cannot get back to +5 V in order to start again. Therefore, the tone is only heard when the OUTPUT port is binary 1.

All we need to do is to alter B0 by changing the number **POKE**'d into the port from an odd to an even number to turn the tone first on, and then off. A program to demonstrate this is shown below.

The Program

```
10  FOR X=1 TO RND*50
20  POKE PORT,1
30  NEXT X
40  POKE PORT,0
50  POKE X=1 TO RND*50
60  NEXT X
70  GOTO 10
```

The program turns the tone on and off for a time determined by a random number between 0 and 50 that is generated by using the command **RND**.

PROJECT 7

A Numeric Keypad

The keyboards used in the T/S 1000 and T/S 1500 consist of a matrix of switches which each connect one ADDRESS line and one data line input. As there are five data inputs (KBD0-KBD4) and eight ADDRESS lines to the keyboard, the maximum number of combinations is forty ($8 \times 5 = 40$) keys.

Unfortunately, the positions of the number keys are not very convenient to use when great accuracy or speed is required. If you use number keys a lot in games or business programs, you might like to build a separate numeric pad. This will enable you to speed up the entry of numbers because you can use the fingers of one hand to "touch" enter numbers without even glancing at the keypad. The T/S 1000 keyboard is made out of three thin pieces of plastic film, so there is very little distance between the top and the end stop of the key movement (0.1 inch). It is, therefore, not easy to tell whether you have pushed the key down far enough to make the switch close. The push keys of the T/S 1500 provide a much more positive feel when pressed.

An excellent keypad can be made from surplus or used calculator or telephone keypads (assuming that the keypad itself is not defective). If a ready-made keyboard is not available, an excellent one can be made from individual momentary contact switches. Select a type with a large flat key surface similar to typewriter keys. It is not worth buying cheap push buttons instead of keys as they only work intermittently; it is

better to have reliable ones. You can then place a piece of paper over each of the keys on which the keyboard symbols can be written. Eleven keys are required, as the numbers 0 to 9 are not very useful if you cannot DELETE any mistakes except by going back to the computer keyboard. The DELETE key requires the pressing of two keys together, 0 and SHIFT. Therefore, the SHIFT key must be included on the numeric pad. Pressing the SHIFT key on its own does nothing, so hitting it accidentally does not give an error on input.

Having the SHIFT key on the numeric pad also means that all the cursor moving keys are also available, SHIFT 5(←), SHIFT 6(↓), SHIFT 7(↑), and SHIFT 8(→). These can be used to EDIT programs quickly, along with the EDIT key which is SHIFT 1. As all of these keys can be reached with one hand if they are grouped in a square, it means that the other hand is free to do other things, such as follow a program in a book or a set of data to be input. This can be very useful, as it is easy to lose one's place when trying to watch the screen and the written program at the same time.

Materials Required

<i>Quantity</i>	<i>Material</i>
11	Keyboard switches
	Project chassis or case, wire, and assorted hardware as required

Construction

The schematic diagram for this project is Figure 7-1. As the keys 1-5, 6-0, and SHIFT are all on different ADDRESS lines, all three must be included on the numeric keypad. These are A8 (SHIFT), A11 (1-5), and A12 (6-0). We also need all of the keyboard data lines (inputs to the computer) KBD0-KBD4.

The keyboard port KBD0-4 is ADDRESSED by the T/S 1000 and T/S 1500 ROM as **INPUT PORT 254** (FE in HEXADECIMAL). But because of the way the computer ADDRESSES its ports, the keyboard port appears at every

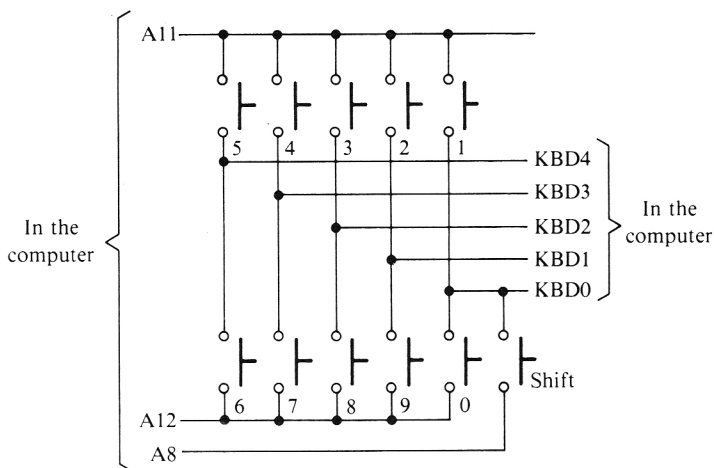


Fig. 7-1

INPUT port ADDRESS. That is when ADDRESS line A0 is at binary 0, the IOREQ and the WR are binary 0.

The upper eight ADDRESS lines (A8–A15) reflect what was in the B register at the time of calling for an input from the port. So the setting of a bit in the B register to binary 0 addresses that key (the ADDRESS line to 0 V) and then looks at the result on the data lines. When a key is pressed, the appropriate data line will also be binary 0. These actions are all done by the BASIC ROM when using **INPUT** or **INKEY\$**. This information has only been included here for the machine code programmer.

We must open up the case of the computer to get at the connections on the printed circuit board inside, and thereby the data lines. If you turn the T/S 1000 upside down, you will see four stuck-on rubber feet. Under three of these feet are screws which need to be removed before the case can be opened. They are under the front two feet and the back left side foot. There are a total of six screws to be removed. All of them need to be removed with a small Phillips-head screwdriver in order not to damage the screw. The T/S 1500 has

five exposed Phillips-head screws that must be removed to open the case.

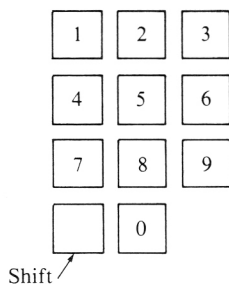
Once the screws in the T/S 1000 have been taken out, the bottom half of the casing can be removed and the printed circuit board can be seen in the top half. This top casing is secured by two more cross-cut screws. By the bottom left-hand side of the printed circuit board you can see the two white plastic strips that connect the computer keyboard to the printed circuit board. These must not be damaged by dropping hot solder on them, so cover them up with a piece of paper. These keyboard strips go into two sockets on the underside of the printed circuit board. The solder strips on the top of the printed circuit board connecting the sockets to the rest of the computer are where we will solder the wires that we will use to attach the numeric keypad.

When the case of the T/S 1500 is opened, the printed circuit board remains attached to the bottom half of the case by a small Phillips-head screw. The two strips will be exposed as described above for the T/S 1000.

The solder connections consist of a group of eight ADDRESS strips and a group of five KBD strips. Soldering onto these strips will not disconnect any of the computer's keyboard functions. None of the wires connecting the computer and the numeric keypad can be over 18 inches long or it causes problems in operating both keyboards. Also make sure that no shorts are made between the strips. See the soldering instructions at the beginning of this book.

A slot must be cut in the bottom left-hand side of the T/S 1000 casing to lead the wires out. This may be done by making two saw cuts, $\frac{1}{2}$ -inch apart, and $\frac{1}{4}$ -inch deep, with a small hacksaw. Then with a pair of pliers, grip the area between the saw cuts and bend the plastic backwards and forwards until the piece breaks off.

The wiring to the keys, in comparison to the computer's wiring, is quite simple. The connections are shown in the circuit diagram. The keys only have two lugs and the order of connection to them is not critical. The physical arrangement of the wires will determine the best order. The ADDRESS lines connect five keys and must be wired from key to key,

**SUGGESTED KEYPAD LAYOUT****Fig. 7-2**

using the wire now attached to the computer. There is only one data line (KBD) to each key and only one ADDRESS line to each key. Only the SHIFT key must be wired to ADDRESS line AB. The keys can be arranged in any order you like, but a somewhat standardized layout, as shown in Figure 7-2, is given as a suggestion.

PROJECT 8

Giant Seven-Segment Display

This is a giant version of a seven-segment display similar to the ones used in pocket calculators and digital clocks. When mounted on a board it gives a display over 1 foot high. It can be used for all sorts of applications and, as it runs from a +12-V supply, may be run off batteries or a power supply. Its attachment to the computer means that an automatic sports-scoring system can be made with a display that can be continuously updated and big enough for everyone to see. Each of the seven elements can be lit separately, so you are not restricted to just the numbers 0 to 9. Many letters of the alphabet can also be displayed. (See if you can list the possible letters.) There are hundreds of combinations of the seven segments available (numbers, letters, and just plain symbols).

Materials Required

<i>Quantity</i>	<i>Material</i>
7	Flashlight bulb, 12 V, 0.1 A
7	Transistor, NPN, 2N697
7	Resistor, 1.2 k Ω
1	OUTPUT port
1	Power supply, +12 V dc, 1 A
	Lucite sheet, mounting board, cardboard tubes, tape, and assorted hardware as required

Construction

Figure 8-1 is the schematic diagram for this project. Seven pieces of clear lucite need to be cut four inches long and $\frac{1}{2}$ -inch wide from a $\frac{1}{4}$ -inch-thick sheet. A special plastic scoring razor blade can be used to cut the plastic to size, or else a saw with a special plastic-cutting blade can be used. Then the $\frac{1}{4}$ -inch edge is glued to the black back board. The rough saw or razor cut will be facing up.

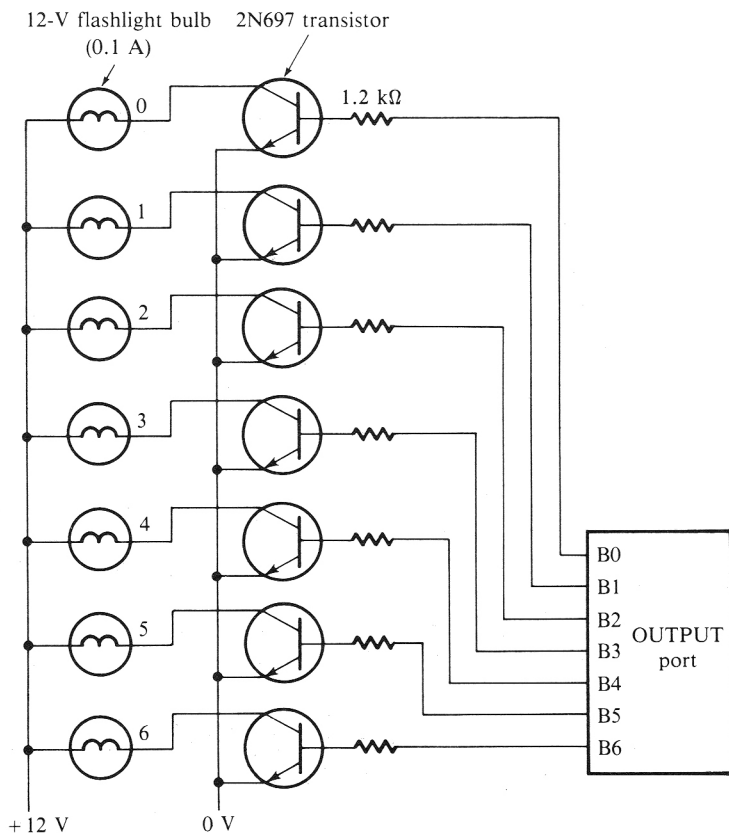


Fig. 8-1

It is necessary to have a rough edge at the top, as we need to diffuse the light through the ridges produced by the cut. Otherwise a lot of the light will be lost bouncing about inside the lucite block off its shiny walls. Next, the six other pieces of lucite are glued onto the board as shown in the diagram (Figure 8-2), leaving plenty of space to fit in the bulbs. The actual size covered by the lucite pieces is up to you, but do not move them too far apart, since this will make the whole display look peculiar.

When lit, the flashlight bulbs send light down the lucite block until it hits an obstruction, then the light diffuses around it, making it glow. When the light hits the far end or sides of the block, most of the light is reflected back into the block again. The only parts of the block that should glow then are the rough cuts at top and bottom. When the block is not lit, the black background shows through.

The bulbs are positioned at the ends of the lucite with the top of the bulb nearest to the end. The rest of the bulb must

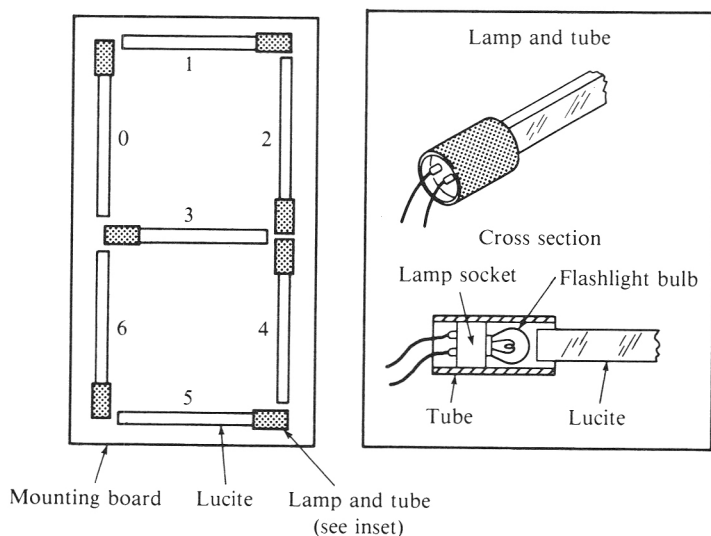


Fig. 8-2

be covered up, so only the glow of the plastic can be seen. Small cardboard tubes, colored black and cut up into one-inch lengths, grip both the bulb and the lucite, while shutting out the light quite effectively. If the bulb tends to slip out, then wind some black tape around the tube with the bulb in it. The tube can be glued to the end of the lucite, if you wish to make it permanent; it still allows you to extract the bulbs when they need changing. The shell (screw) terminal of the bulb should be connected to all the other bulb screws. Then only one wire is required to go to the positive terminal of the +12-V supply. The center terminal of the bulb should be wired to the collector (C) of its own transistor. The length of the wires connecting the transistors and the display can be quite long, provided the resistance of the wires does not excessively drop the voltage across the bulb. The longer the wires, the greater the resistance.

The wiring can be made neat by drilling holes in the board and taking all the wiring to the back of the display, using the tubes to cover up the holes.

The transistors have metal cases, which are connected to the collector, so the cases must not touch each other. Also, do not allow any bare wires near them, as contact with other wires can destroy the transistors. Always leave plenty of length on the leads of the transistor and, before soldering them, cover them with insulated tubing. When soldering, always use a heat sink. This can be just a pair of pliers gripping the metal of the leads between the soldering iron and the transistor.

All of the transistors' emitters (E) are connected to the 0-V line (this must also be connected to the computer's 0-V line). The collectors are all connected via their flashlight bulb to the +12-V line. The transistors are, therefore, only being used as a switch to turn the bulbs on and off. When the port bit goes to binary 1, the voltage at that bit is +5 V (because it is connected to the +5-V line and not the +12-V line). This means that the voltage difference between the base (B) and the emitter (E), which is at 0 V, is +5 V. Now a transistor's resistance (between the collector and emitter) varies from very high to very low depending upon what the base voltage is. If

the base is connected to 0 V, the resistance (from the collector to emitter) is so high and the current flowing through it so low, that the bulb will not light. If, however, it has a voltage of only +0.6 V on the base, it starts to drop its resistance. If +5 V is applied to the base, then the resistance is so low that it will pass all of the current that the bulb needs to light (almost 0 ohm). The transistor is turned off by setting the port bit to binary 0 (0 V). A transistor is used because the flashlight bulb needs 0.1 A to glow brightly enough to be seen properly. If you used an IC to switch this current, it would soon overheat and blow up. The transistor 2N697 can take up to 0.5 A safely without overheating. The transistor should only be used on a dc supply and must be connected correctly or the transistor will be destroyed. The 1.2-k Ω resistors protect the transistor and the port by limiting the amount of current flowing between them.

Now that we have built the display, we must have a program to run it.

The Program

```

1  REM 000000000000
2  FOR A=0 TO 9
3  INPUT B
4  POKE 16514+A,B
5  NEXT A
6  INPUT A
7  POKE PORT, PEEK (16514+A)
8  GOTO 6
10 FOR A=0 TO 127
20 POKE PORT,A
30 INPUT A$
40 IF A$ <> "" THEN PRINT A;A$
50 NEXT A

```

First replace **PORT** in lines 7 and 20 with the location of your memory mapped **OUTPUT** port. If you use a machine code routine to write to your port, then substitute the correct **POKE** followed by the **USR** routine.

Now **RUN** 10 and you will see zero output to the display (all the lamps should be off). Press **ENTER** and the next number will be **POKE**'d into the port. If you like this combination, you can store its number on the screen by pressing another key before **ENTER**. I suggest that this other key be the letter or number you think it looks like. The number of the combination is then printed on the screen, followed by the key (or keys) that you pressed before **ENTER**. The display now shows the next combination.

The second program starts by pressing **RUN** 1, then **ENTER**. The following numbers should then be put in:

123
10
55
31
78
93
125
11
127
79

Press **ENTER** after each number is typed in. This **POKEs** these numbers into the first line which is a **REM** containing ten zeros. The first 0 is location 16514 and has 123 **POKE**'d into it. These numbers are the combinations required to produce the digits 0 to 9 in the correct order, so **PEEK** 16514 + 9 = 9. Lines 6 to 8 ask for a number to be **INPUT** (from 0 to 9), and then **POKEs** it on to the display. It then waits for the next number.

One word of warning: the number 126 (HEX 7E) cannot be put in a **REM** line, as it signals the end of a **REM** line and would corrupt the listing.

PROJECT 9

Scoreboard

The purpose of this project is to show how the T/S 1000 and T/S 1500 can make decisions on external events. The inputs to the computer are three switches as shown in Figure 9-1. Two are push-to-make and one is a locking type. They are connected to the bottom three bits of the INPUT port (B0-B2) and thus give a combination of values between 0 and 7. As the rest of the inputs are not connected, they are all set by the port to binary 1, giving a count of 248 which should be added to the result of the other bits. Therefore, the count can vary between 248 and 255.

These switches could be used to monitor such events as a door opening, the level of water in a tank, or even the number of cans of beans moving along a conveyor belt. But in this case, we will use it to monitor a scorer who wants to be

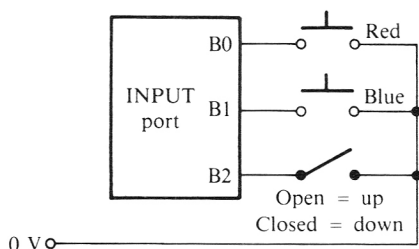


Fig. 9-1

able to increase or decrease the score of either of two teams by one. The scorer wants no mistakes made if two keys are pressed by mistake. The result is displayed on the screen to be checked.

Materials Required

<i>Quantity</i>	<i>Material</i>
2	Switch, momentary contact
1	Toggle switch, single-pole single-throw
1	INPUT port
	Project case and assorted hardware as required

There are three keys available to the scorer (Figure 9-2). One changes the score, either incrementing (adding one) or decrementing (subtracting one) and back again. Just operating this switch on its own does nothing to the score. The other two (one for each team) tell the computer when to increment or decrement the score of that team. Pressing them simultaneously is obviously wrong from the scorer's point of view, so the computer must stop this from altering the score should it happen accidentally.

Each switch, when operated, causes that bit to change to binary 0 by connecting it to the 0-V line of the computer. If any of the switches are left open then, that bit is held to binary 1 (+5 V) by a resistor connected inside the port. Therefore, if the red count is to be incremented, the input should be 254 in decimal, with only the red switch operated

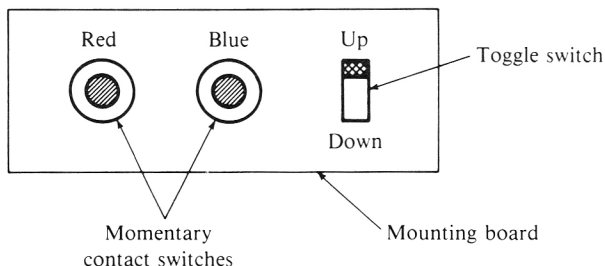


Fig. 9-2

(11111110). If the blue count is to be decremented, then the blue and the UP/DOWN switches are operated, giving a result of 249 (11111001). If the red and blue switches are not pressed, the result is 251 (if the UP/DOWN switch is operated) and 255 if no switches are operated. Line 60 in the program sends the computer back to have another look at the switches, if no red or blue switch has been pressed. When both red and blue switches are pressed the score is not affected because of the absence of an **IF** line that would react to this number. To use the program, replace **PORT** with the location of your memory mapped port or **USR** routine for your **INPUT/OUTPUT** port.

There is nothing to stop you from changing the program to read the results from up to eight switches, one switch to each bit, or checking for more combinations from the switches given. By giving the switches another use, the display could show you how many cans of beans you use in a week!

The Program

```

10  LET B=0
20  LET C=0
30  PRINT AT 0,0; "RED"; TAB 7; "SCORE",
    "BLUE"
40  GOTO 100
50  LET A=PEEK PORT
60  IF A=255 OR A=251 THEN GOTO 50
70  IF A=250 THEN LET B=B-1
80  IF A=249 THEN LET C=C-1
90  IF A=254 THEN LET B=B+1
100 IF A=253 THEN LET C=C+1
110 PRINT AT 1,1; B; " ";TAB 18; C, " "
120 GOTO 50

```

PROJECT 10

Wheel of Fortune

This project produces a computerized version of the spinning wheel of fortune seen at fairs and casinos. Instead of a pointer, we shall use 16 light-emitting diodes (LEDs) which light up one at a time around the circle.

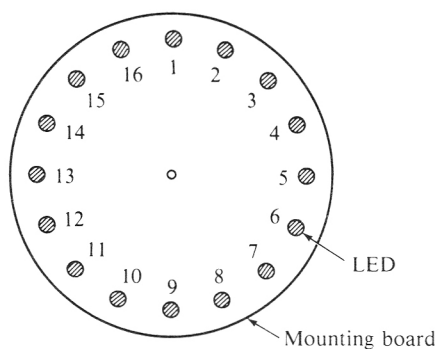
The T/S 1000 and T/S 1500 can output a number of rotations, before counting up to a number chosen at random, using the **RND** command on the last turn. The winner, of course, is the person who has guessed the number where the count stops. With 16K RAM the T/S 1500 can even print the name of the winner on the screen.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	OUTPUT port
4	IC 7400 or 74LS00 (IC 1-IC 4)
1	IC 74LS154 (IC 5)
16	LED
16	Resistor, 220 Ω , 1/4 W
1	Capacitor, 0.1 μ F, 60 V
1	Power supply, +5 V dc, 500 mA
	Mounting board, assorted hardware as required

Construction

As shown in Figure 10-1 we need to draw a large circle for which a string and a pencil can be very useful. Tie the string around the center of the pencil and mark on the string the radius you want to draw on the board. This will be the display of your wheel of fortune. Holding the string in place where

**Fig. 10-1**

the center of the circle will be, rotate the pencil around in a half circle. Then change the hand holding the string and draw the other half circle. Draw a total of three concentric circles, one for the outside, one for the LEDs ($\frac{1}{2}$ inch in from the first), and one for the numbers from 1 to 16. Drill 16 equally spaced $\frac{1}{4}$ -inch holes around the second circle. The board is now ready for painting your design before inserting the LEDs.

When the board is dry, insert the LEDs into the holes with the anode lead pointing away from the center of the circle. This is the shortest of the two leads. To the cathode, now pointing inwards, solder one end of the 220- Ω resistor. The other end of the resistor should be soldered to a circle of wire connecting all similar ends of the resistors.

Now wire up the ICs to the OUTPUT port on a piece of board near the port (Figure 10-2). The ICs need to be wired up in the correct sequence around the circle, so check the pin layout in the diagram. Make sure that the address lines of the 74LS154 are connected to the correct bits of the OUTPUT port. The wires from the LEDs go to the outputs of the NAND gates in ICs 1 through 4 and the wire circle on the display goes to the 0-V line. There are four NAND gates in each IC and the right LED must go to the right gate output. The output of the gate connected to the NOT 0 ($\bar{0}$) output of IC 5 must go to LED number 1, output NOT 1 ($\bar{1}$) to LED

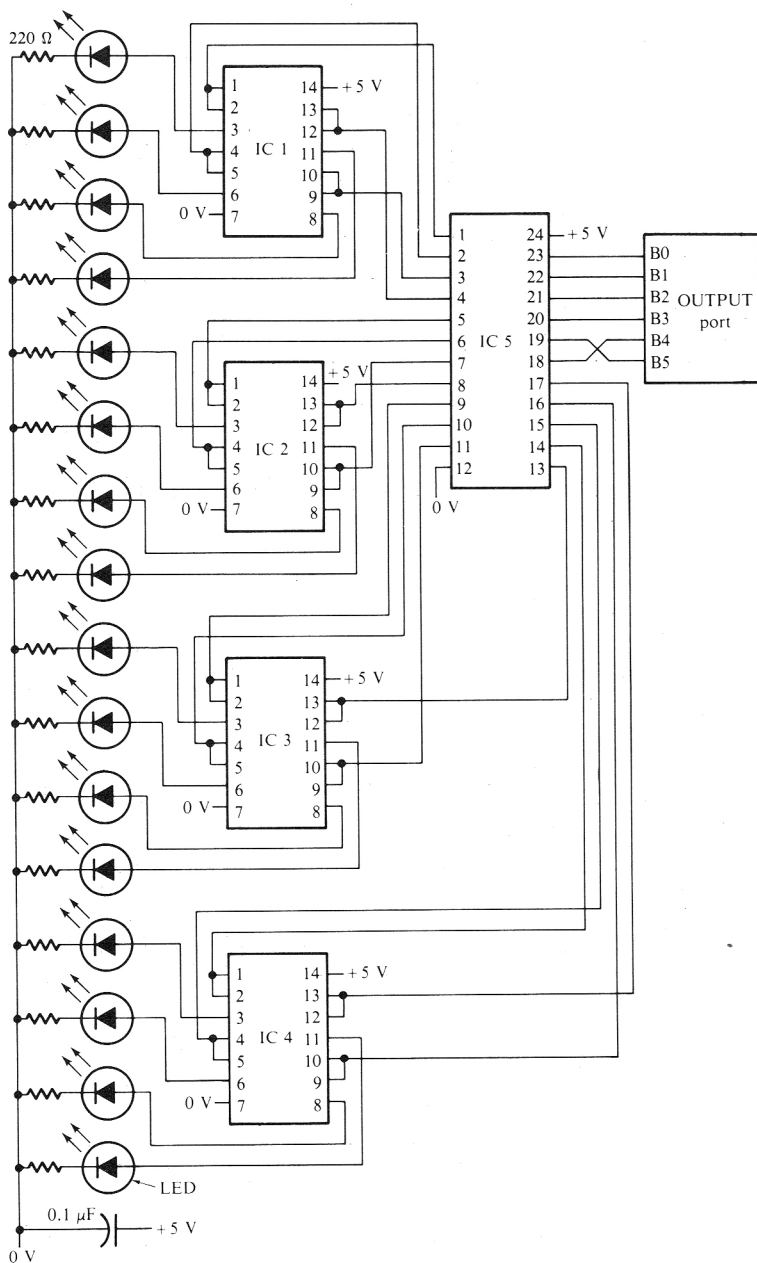


Fig. 10-2

number 2, and so on. The 0.1- μ F capacitor must be fitted on the IC board between the supply lines. Don't forget to connect the 0-V rail to the computer's own 0-V line, otherwise there will be no communication between the port and the circuit.

How It Works

Most of the work is done for us by the 74LS154 (IC 5). This IC will decode a binary input of 0 to 15 into 16 individual outputs. Now a binary input of 15 can be expressed as only four bits and this is how the number is output by the port on bits B0-B3. The computer does the conversion from decimal to binary before **POKE**ing the number into the port. If you are not sure about binary, read the chapter called "Getting In and Out of the Computer" again.

The output consists of a single pin that will go to binary 0 only when the correct binary number is input to the IC. This is why it is called a NOT output, meaning that if 0 is input (all the pins A0-A3 are binary 0), then the 0 pin will go to binary 0. In other words, the decoded output goes down to binary 0, not up to the usual binary 1. So the output is inverted to normal and hence the bar above the number meaning NOT.

Each output has a 7400 IC gate acting as source of current for the LED, because if it were attached directly to IC 5 it might overheat. This has been used many times in this book, so for a further explanation of how this works, see the explanation in the Monitor project (Project 3).

The two lines connected to the OUTPUT port bits B4 and B5 ($\overline{E0}$ and $\overline{E1}$) must be at binary 0 in order for the 74LS154 to work, otherwise IC 5 ignores any inputs at all.

The internal workings of the 74LS154 consist of 16 NAND gates, one for each output. A NAND gate output will change from binary 1 to binary 0 only when all the inputs are binary 1.

Four of the inputs are driven from the ADDRESS lines (A0-A3), either directly operating the gate to a binary 1 input or through an inverter included in the chip, which means the

gate operates to binary 0. Each gate has different combinations of direct and inverted ADDRESS lines, thus all 16 combinations are covered.

One of the inputs is driven from a combination of the inputs $\overline{E0}$ and $\overline{E1}$, which will stop all of the gates from going to an output of binary 0 by keeping one of the inputs at binary 0.

The main use of this IC is to ADDRESS 16 different locations in memory (just as our LEDs are operated), by turning on the appropriate chip only when the correct binary number is input. The INPUT lines need not all be from the ADDRESS lines (A0–A15 on the T/S 1000). To have a memory location addressed, three signals are required, $\overline{MREQ}$ (NOT Memory REQuest), $\overline{RFSH}$ (NOT a ReFreSH ADDRESS, used on the T/S 1000 to output the character set) and the ADDRESS lines. The $\overline{MREQ}$ must be binary 0, the $\overline{RFSH}$ must be binary 1, and the ADDRESS must be correct in order to turn on the chip at the correct time. The IC 5 can cope with all of this; $\overline{MREQ}$ operates both $\overline{E0}$ and $\overline{E1}$. The ADDRESS consists of $\overline{RFSH}$ connected to A3, and the ADDRESS lines A0–A2 connected to the same lines on the T/S 1000. This means we can only use the top eight outputs, as the A3 line is connected to the $\overline{RFSH}$ and *only* when this is binary 1 will it be right. Therefore, the ADDRESS outputs $\overline{8}$ to $\overline{15}$ will be equal to the ADDRESS input on the other lines A0–A2. But there is a problem. We have left out ADDRESS lines A3–A15 on the T/S 1000; therefore, the ADDRESS lines are correct every count of eight. We must make use of another chip to cope with A3–A15 so that, when they are all binary 0, the input to $\overline{E1}$ is binary 0. So now $\overline{MREQ}$ only operates $\overline{E0}$, and $\overline{E1}$ is operated by all the ADDRESS lines because it is in binary 0, thus using the outputs from IC 5 to operate chips at ADDRESSES A0 to A8 from outputs $\overline{8}$ – $\overline{15}$. This is called *ADDRESS decoding*.

Not to worry, we only have to cope with six inputs and we can output all of them from the OUTPUT port. We need to **POKE** the correct number at the correct port and the computer will do the rest. Here is a program to demonstrate that fact.

The Program

```

10 LET A$="1111111111111111"
20 PRINT "NUMBERS LEFT"
30 GOSUB VAL "500"
40 INPUT B
50 IF B<VAL "1" OR B>LEN A$ OR A$(B)="1"
   THEN GOTO 40
60 LET A$(B)="0"
70 GOSUB VAL "500"
80 IF C=VAL "1" THEN GOTO 40
90 CLS
100 LET B=VAL "16"
110 FOR A=VAL "1" TO VAL "5"
120 GOSUB VAL "600"
130 NEXT A
140 LET B=INT(RND*15)
150 PRINT "THE WINNER IS";B+1
160 STOP
500 LET C=VAL "0"
510 FOR B=VAL "1" TO VAL "16"
520 PRINT AT B,10;B
530 IF A$(B)="0" THEN PRINT AT B, 10; " ■ ■ "
540 IF A$(B)="1" THEN LET C=1
550 NEXT B
560 RETURN
600 FOR X=C TO B
610 POKE PORT,X
620 FOR N=C TO VAL "200"
630 NEXT N

```

640 **NEXT X**

650 **RETURN**

One thing about this program is that we have had to use **VAL** "1", instead of 1 and so on, in order to get all of it to fit into a 2K T/S 1000. The record of numbers that have been sold is kept in a string A\$. Every time one is sold, check to see that it makes the character in A\$ equal to 0, then **INPUT B**. Subroutine 500 prints the numbers not sold down the center of the screen, while variable C keeps an eye open to see if all the numbers have been used. When all the numbers have been used, the LEDs are illuminated in rotation five times before they stop on the random number. All of the program must be run in the **FAST** mode to speed things up.

If you are using a T/S 1500 or if you have a 16K RAM pack with the T/S 1000 add the following lines to also record the name of the winner:

```
5  DIM B4(16,32)
45 INPUT B$
145 PRINT B$(B + 1)
```


PROJECT 11

Analog-to-Digital Converter

Commonly called an A/D, this device is very useful for anything that needs measuring. In fact, it is so useful that we shall be using it as a base for other projects within this book.

This device allows us to input a level to be measured at one end and arrive at a number at the other end that can be read by the computer. The level input will vary the number given out. The speed at which this can be done is quite fast when using a machine code routine (greater than 50 ms). It can be done using a BASIC **FOR/NEXT** loop to count the number, but the time taken could be up to 5 seconds for each measurement. Using the machine code routine given, it is just as fast as many commercial A/D converters, but so much cheaper!

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Resistor, 1.2 k Ω , 1/4 W
1	Resistor, 100 k Ω , 1/4 W
1	Capacitor, 0.5 μ F
1	IC NE555
1	INPUT/OUTPUT port
1	Regulated power supply, +5 V dc, 200 mA
	Project case and assorted hardware as required

Construction

The schematic diagram for this project is shown in Figure 11-1. All the joints in this project must be well made because the measurements taken need to be as accurate as possible. As an example, in the program given the only inaccuracy in the number returned is caused by a change in air temperature. A change of just one-third of a degree celsius can alter the results by ± 1 . So unless the joints are very good, their resistance will vary and give the wrong reading.

All the wiring for the IC should be made to an IC socket and the IC should be inserted in the socket only after all the connections to the socket have been checked. There are only eight pins on the NE555, four per side; two of these are left unconnected (pins 4 and 5). The wiring should be kept as short as possible and the board containing the A/D converter should be as close to the port as possible. A +5-V regulated supply must be used, to prevent voltage changes affecting the measurements from one reading to another.

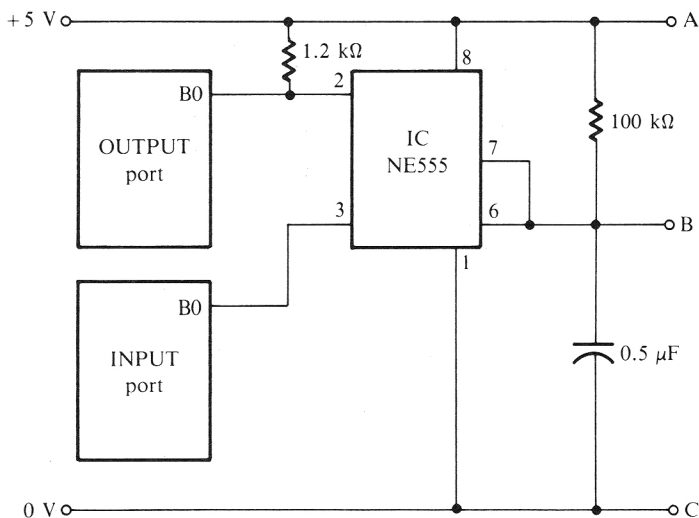


Fig. 11-1

How It Works

We convert the level to a number, using a two-stage process.

1. Use the level to control the output of the NE555. The level input, determining how long the output is, should be at binary 1 (+5 V). The higher the level the shorter the pulse.
2. Measure the length of the pulse. This is done by using a machine code loop to count the number of times the NE555 has set the INPUT port to binary 1 (after being started off by a change in the OUTPUT port). The longer the time before the INPUT port returns to binary 0, the larger the number. A preset limit is used so that the 50-ms limit is not exceeded.

The whole project depends on controlling the time of this pulse.

The NE555 is a comparator IC: it takes an input from pin 6 and compares it to two-thirds of supply voltage (in this case +5 V). When triggered with pin 6 at two-thirds the supply voltage, the IC resets itself. This resetting consists of connecting pin 7 to 0 V and keeping it there until the IC is triggered again. When triggered, the output on pin 3 is +5 V (binary 1) until the IC is reset, then it returns to its normal state of 0 V (binary 0).

Pin 2 is the trigger input and starts the IC off by going from +5 V (binary 1) to 0 V (binary 0) and back to +5 V (binary 1).

When triggered by the OUTPUT port going from binary 1 to binary 0 and back to binary 1 (operating pin 2), the IC first changes pin 3 to +5 V (which is connected to the INPUT port, B0), and then causes that bit to be set to binary 1. The 0 V is removed from pin 7 and the capacitor now starts to charge via the 100-k Ω resistor.

After a period of time (set by the capacitor and resistor), the voltage rises to two-thirds of the supply voltage. As this is connected to pin 6, this resets the IC. The output changes

to binary 0, and 0 V is put on pin 7 until another trigger pulse is received. Because pin 7 and pin 6 are connected, the ends of the capacitor are shorted and the capacitor discharges. It will remain this way until pin 2 is triggered again and removes the short.

If a binary 0 is output on B0 of the port and then restored to binary 1, this starts the timing under control of the resistor/capacitor. Binary 1 is input on B0 of the port until the timing finishes. If we alter the value of the resistance by putting another resistor across it, it will change the time and, therefore, the number returned to BASIC.

The program given is in two parts. The first part (below) is in BASIC to insert the machine code into a **REM** statement at the first line of the program. In the second part (see p. 57), the machine code routine checks the input from the port after the output has triggered the IC, and returns when it returns to 0. The register BC counts the number of times the routine loops through checking the input bit. This is the number returned from the machine code routine to BASIC. The machine code routine must be run in the **FAST** mode or the time counted will vary due to the **SLOW** mode interrupting the processor in order to output the screen. The **SLOW** mode must be restored if the screen is used without input, thereby stopping the program to display the screen.

This routine is entirely relocatable and in the BASIC program is stored at location 16514, the first byte after the **REM** byte in the first line of the program.

The Program

```

10  REM MACHINE CODE ROUTINE
20  REM SET UP PORT IF NECESSARY
100 REM ENTER MACHINE CODE
110 LET X=16514
120 PRINT "ENTER EACH NUMBER THEN PRESS
      ENTER"
```

```
130 PRINT "33 001 0062 255 112 119 190", "192 3  
24 251"  
140 LET N=0  
150 INPUT A  
160 POKE X+N,A  
170 IF A=251 THEN GOTO 200  
180 LET N=N+1  
190 GOTO 150  
200 LET A=PORT LOCATION  
210 LET B=INT(A/256)  
220 LET C=A-B*256  
230 POKE X+1,C  
240 POKE X+2,B  
250 FAST  
260 LET C=USR X  
270 SLOW  
280 PRINT C  
290 INPUT A$  
300 CLS  
310 GOTO 250
```

RUN the program, after changing line 110 to the location of the machine code routine, if necessary, and changing the port location to a number.

Line 10 has enough bytes to contain the machine code routine, one letter for each byte. If it needs to be longer, just add more letters to line 10 before running the program. Line 260 calls the machine code routine in line 10 via the **USR** command at location X. If the port requires the use of a machine code routine, then change the amount and the string of numbers in line 130. The result printed will be the time taken.

Machine Code Routine for Collecting Number

<i>Instruction</i>	<i>Hex</i>	<i>Decimal</i>	<i>Comment</i>
LD HL, PORT LOCATION	21 xx xx	33 xx xx	Ignore if your port uses a machine code routine.
LD BC,0	01 00 00	1 0 0	Start count at 0.
LD A,255	3E FF	62 255	The value that the port should be at if no other bits are used.
LD(HL),B	70	112	Set bit low (call machine code routine if necessary).
LD(HL),A	77	119	Set bit high (call machine routine if necessary).
CP(HL)	BE	190	Compares port to A register (call routine to place value of port into D register, then CP,D). A register is not changed, only flags are changed.
RETNZ	C0	192	RETURN if A not same as port.
INC BC	03	3	INCrease count by 1 as bit is still binary 1.
JR CP(HL)	18 FB	24 251	This needs to be altered if calling machine code routines to compensate.

PROJECT 12

Light Pen

A light pen can be very useful for many purposes, from reading bar codes printed on boxes and cans in the supermarket, to reading options from the TV screen by changing spots on the screen from black to white. A bar code is the set of black lines on a white background that you find on the side of, say, a box of cornflakes. It is a numeric code based on the thickness of the lines. The light is reflected from the white and absorbed by the black lines.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Photo-transistor, FPT100
1	A/D converter
	Ball-point or felt-tip pen case, paper fasteners, and assorted parts as required

Construction

The most important part of this project is its case. An old felt-tip or clear ball-point pen case is required, as we are going to fit into it the photo-transistor which is shown in the circuit diagram (Figure 12-1).

The end and ink refill should be removed so that the photo-transistor can slide up into the middle of the pen (Figure 12-2). Although I have specified a FPT100 photo-trans-

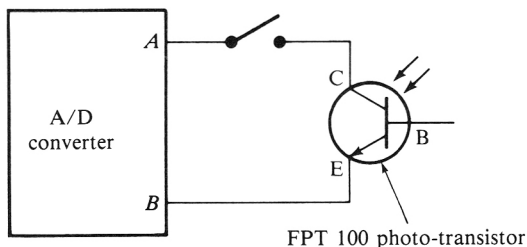


Fig. 12-1

sistor, other types can be tried. Drill two $\frac{1}{8}$ -inch holes in one side of the pen case at least $\frac{1}{2}$ inch apart. Then cut down one side of each brass paper fastener to $\frac{1}{4}$ inch. Solder a wire onto the long strip of a paper fastener and solder another insulated wire onto the short strip of another paper fastener. Push the insulated wire that is attached to the short strip down the hole farthest away from where the photo-transistor is going to be put. Push this wire up toward the top of the pen, pulling the paper fastener along after it; make sure that the long strip remains on the outside of the pen and points toward the pen's tip. The other wire (attached to the long strip) should be pushed down toward the tip of the pen while pulling both

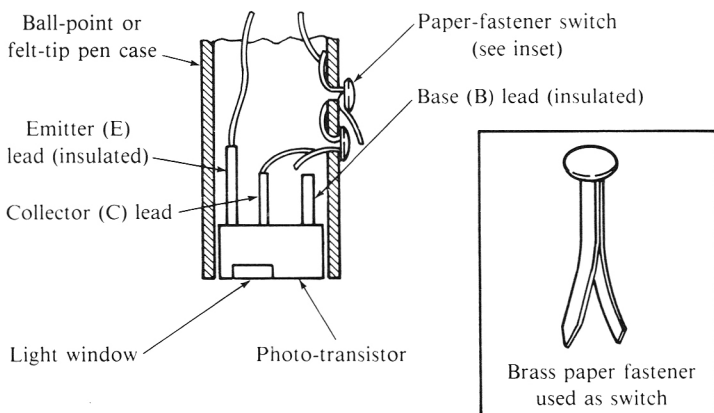


Fig. 12-2

strips of its paper fastener down inside the pen after it. The two paper fasteners form the switch in the circuit diagram, which should be pressed every time a reading is required.

The base of the photo-transistor is covered with plastic tubing (called *spaghetti*). Heat-shrinkable tubing works fine in this case. The wire from the long paper fastener strip has plastic tubing covering it. This tubing should be long enough to cover the entire transistor collector. Bare the end of the wire and solder it to the collector using a heat sink between the transistor and the soldering iron. Next, pull the tubing over the bare collector wire. Push a new piece of wire down the pen from the top to the tip and push tubing over the wire. The tubing should be long enough to cover up the emitter wire. As before, solder the bared wire to the emitter wire and push the tubing over it. Push the transistor into the tip of the pen, pulling the wire attached to the emitter out of the top of the pen as you go.

The two wires coming out of the pen should be long enough to connect up to the A/D converter project. The emitter wire must go to the connection marked B or it will not work. To use the pen, press the brass strip onto the brass button of the other paper fastener when you are trying to take a reading.

How It Works

The window in the front of the transistor lets the light in and alters the resistance between the emitter (E) and the collector (C). This is placed across the 100-k Ω timing resistor by connecting it to the A/D converter's terminals A and B.

The emitter must be connected to terminal B because the current will flow only through the transistor from collector to emitter and terminal A is connected to the +5-V line. The base is left unconnected, as only the light alters the resistance of the transistor.

The resistance can vary between 0 and 2500 Ω with the higher number being the darkest. This means that the difference can be detected between black and white squares on a page lit by daylight only.

I leave the programs that will use this light pen to you, since the number from the A/D converter's machine code routine can be checked to see if it is within the limits set by the program. The sensitivity of the light pen is determined by the resistance which must be within the limits of the program.

PROJECT 13

A Shift Lock for Keyboards

This project provides a shift lock for single contact keyboards, such as the T/S 1000. It will enable the user to hold the SHIFT contact closed until another momentary switch is operated. This means that two key operations can be done with one finger. For instance, to DELETE, press the SHIFT LOCK switch and release it; then press the 0 key. This will delete the letter or number to the left of the cursor. To be able to type in nonshifted letters and numbers, the UNLOCK key must be pressed to restore the keyboard to normal.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Opto-isolator, TIL111
1	IC 7400 NAND gate
1	Zener diode, 5 V, 400 mW
1	Resistor, 390 Ω , 1/4 W
1	Resistor, 1.2 k Ω , 1/4 W
2	Switch, momentary contact
1	Switch, single-pole single-throw
1	Battery, 9 V
1	Battery connector for 9-V battery
	Project case and assorted hardware as required

Construction

Figure 13-1 is the schematic diagram for this project. The whole unit can be built into a separate board with the LOCK

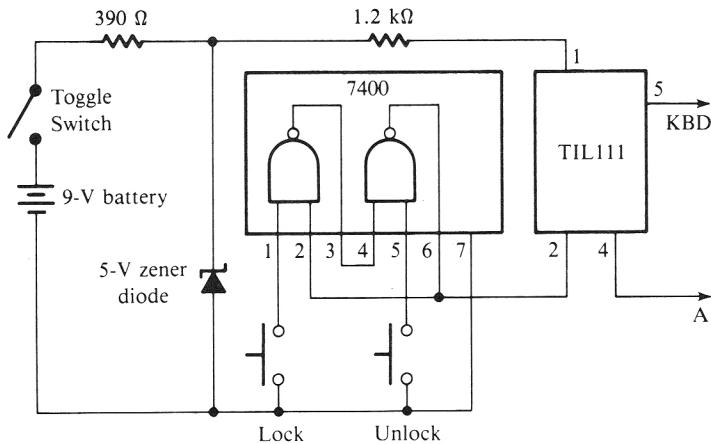
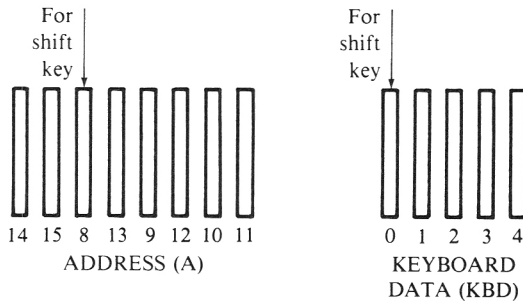


Fig. 13-1

and UNLOCK switches on top. The ON switch is a locking type and is included so that the unit can be switched off when not in use (to save battery power); it can be omitted, if you can easily disconnect the battery every time you have finished using it. One reason for using a battery is that it does not place any strain on the internal regulator of the T/S 1000; but more importantly it means that the only connections needed to the computer are the keyboard socket connections on the printed circuit board. Therefore, no damage can be done to the computer by shorting out any connections. The 7400 IC used has four NAND gates in the chip, but only two are required for this project, so connections 8–13 are left unconnected. Another lock could be constructed using this half of the IC.

The connections to the T/S 1000 are made by soldering wires from the bottom of the computer's own keyboard sockets (AB and KBD0), to the printed circuit board (Figure 13-2). The TIL111 must be connected correctly, otherwise no contact will be made. The correct connections are ADDRESS line to pin 4, and KBD line to pin 5. This is because it acts like a diode and only conducts current one way, from KBD to A. The circuitry draws only 10 mA so the battery will last



T/S 1000 AND T/S 1500 KEYBOARD STRIPS

Fig. 13-2

some time. As only spring-loaded keys are usually included on an external keyboard, any spare keys may be used as the LOCK and UNLOCK keys. The + end of the zener diode is usually indicated by a band.

How It Works

The device used here is called a latch and is used to short out the SHIFT key on the computer; it is released only when another key is pressed. The latch is made from a 7400 IC and once operated, it will not be released by the same key.

When switched ON, the inputs to NAND gates are held to binary 1 by internal resistors. So pins 1 and 2 are at binary 1, this operates the NAND gate and it drops its output to binary 0. This stops the second gate from operating, as its input pin 4 is connected to the output of the first gate. A NAND gate must have binary 1 on both inputs to operate. This means that there is binary 1 output by the second gate and that stops the LED from lighting up because it needs a voltage difference of at least 1.2 V to operate. The LED has +5 V at both ends at this stage. As pin 6, the output of the second NAND gate, is connected to the input of the first gate, the first gate operates as long as the LOCK key is not pressed.

Pressing the LOCK key connects pin 1 to binary 0 (0 V) and releases the NAND gate, which now has only one binary 1 input (pin 2). The output, therefore, changes to binary 1 and operates the second gate since pin 5 is also connected to

binary 1. This drops the output of the second gate to binary 0, lighting the LED in the TIL111. The lighting of the LED shines light on the photo-transistor (also included in the same IC), which turns fully ON (just like a switch), so that 500 Ω appears between the COLLECTOR (pin 5) and the EMITTER (pin 4). This acts just like operating the SHIFT key on the T/S 1000.

The output of the second gate is also connected to the input pin of the first gate, so this also changes to binary 0, which stops any change in the input pin 1 from having any effect. The NAND gate will always output binary 1 if either of its inputs are binary 0. So the output is said to be latched, as it cannot be changed by releasing the key that once operated it.

When the UNLOCK key is pressed, it connects the input of the second gate to binary 0, thereby releasing the gate and changing the output (pin 6) to binary 1. This turns off the LED inside the TIL111 and removes the contact between pin 5 and pin 4. The photo-transistor now gives a resistance of many million ohms and the computer believes the keyboard has released the SHIFT key. The output of the second gate is now binary 1 and, as it is connected to input pin 2, it turns on the first gate because the other input (pin 1) is also connected to binary 1 (via the IC). This changes the first gate's output to binary 0, and stops the second gate from changing its output due to a change in the UNLOCK key. Thus, it is now latched into the binary 1 output until the LOCK key is pressed again.

The 1.2-k Ω resistor is used to limit the current that flows through the LED and the second gate in order to prevent possible damage. The zener diode maintains a constant +5 V across its terminals by altering its resistance up or down, until it reaches +5 V. The excess voltage of 4 V is dropped across the 390- Ω resistor. If you want to supply enough current to operate an LED, then the value of the resistor will have to be reduced. The total current taken by this device is only 10 mA, so it can be operated from any 9-V battery.

PROJECT 14

An LED Thermometer

This is a very inexpensive thermometer to build since it uses only two yellow LEDs as its sensor. When the computer is connected to the A/D and this thermometer, you will have a very sensitive instrument with approximately six places per degree celsius.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Resistor, 1.2 k Ω , 1/4 W
2	LED, yellow
1	A/D converter
	Project case and assorted hardware as required

Construction

The schematic diagram for this project is shown in Figure 14-1. The variation in temperature is doubled by using two LEDs mounted next to each other. The 1.2-k Ω resistor should be mounted next to the A/D converter, so that its temperature is unaffected by temperature changes of the sensor.

The LEDs, if expected to measure air temperature, should be well insulated from any metal or this thermometer would measure the temperature of the metal instead. If you want to use the thermometer in a corrosive atmosphere or liquid, such as an aquarium, mount it inside a test tube sealed with a rubber stopper. The PVC covering of the cables should be epoxy

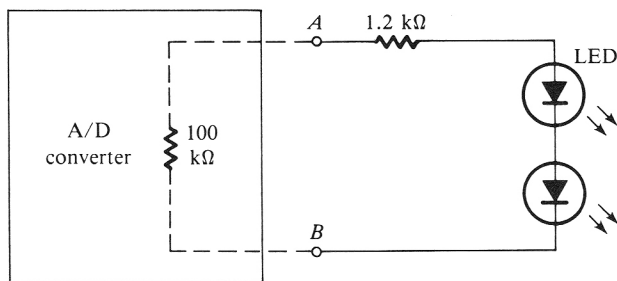


Fig. 14-1

resined to the stopper. One advantage of this thermometer is that there is no line voltage connected to the sensor and, even if you broke the protective glass tube, the water would only disable the thermometer until the LEDs were repaired. If this happened in an aquarium, the only thing that could hurt the fish would be rust. It can also be used safely in an explosive atmosphere because all of the glowing elements are sealed in plastic!

No damage to the power supply or A/D can occur. The circuit is protected by limiting the current fed to the LEDs by using a 1.2-kΩ resistor. The anode of the first LED is connected to the 1.2-kΩ resistor, and its cathode is connected to the other LED. The cathode of the second LED is connected to the wire to point B on the A/D converter. The cathode of an LED is easily identifiable since it is the longer lead of the LED.

Both LEDs glow during normal operation, indicating that the circuit is in working order. Any break in the wiring or shorts in the cabling cause the LEDs to go out. Therefore, the LEDs indicate any fault in the sensor network.

How It Works

The resistance of a diode through which current is passing varies very slightly with temperature (Figure 14-2). We can, therefore, get a condition called *thermal runaway*; a point

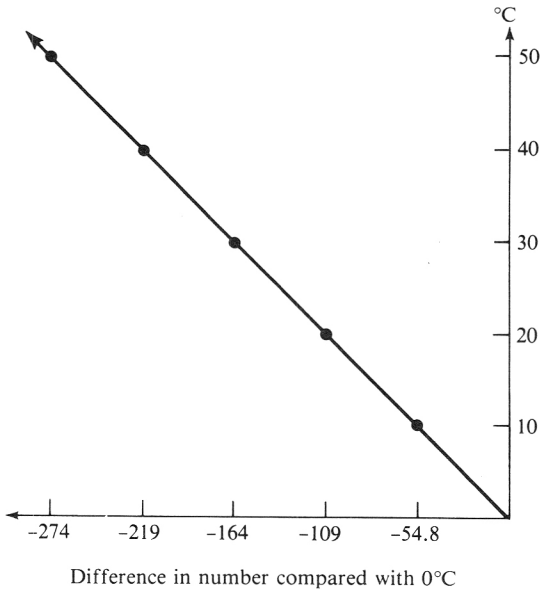


Fig. 14-2

which is reached when some of the current causes heat to be created inside the diode. This current causes a rise in temperature that drops the resistance of the diode. The drop in resistance means that more current can flow, creating more heat inside the diode. This accelerates until the diode melts due to the heat created within it.

We don't want either of the diodes (LEDs) to melt, so we have included a 1.2-k Ω resistor to limit the current flowing through them. This current must be enough to ensure that the LEDs conduct the power (indicated by their glowing), but not enough to melt them. It is not sensible to use this form of thermometer for any temperatures in excess of 50°C. The resistor is also mounted out of harm's way (next to the A/D converter), so that if anything should happen to cause a large drop in resistance, the resistor can keep the current down to a safe level, that is, one that the A/D converter can easily handle.

A Program for an Aquarium Thermometer

```
10 REM MACHINE CODE ROUTINE
20 REM SET UP PORT IF NECESSARY
30 LET A=USR 16514
40 IF A>MAX THEN POKE PORT, 2
50 IF A<MIN THEN POKE PORT, 0
60 IF A=0 OR A=DANGER THEN LET A=0
70 IF NOT A = 0 THEN GOTO 30
80 POKE PORT, 128
90 PRINT "CIRCUIT FAILURE"
```

Remember to change PORT for the OUTPUT port location or call the appropriate machine code routine. Change MAX and MIN to the numbers that correspond to the temperature range that you want to use. The danger level is the maximum number expected from the A/D converter. The PORT is arranged so that B0 operates the A/D, and B1 operates a heater switch (binary 1 on B1 turns it on, and binary 0 turns it off). B7 of the OUTPUT port is used to turn on an alarm of some sort (perhaps the Beeper project, Project 6) by the bit going to binary 1. If the number returned by the routine is too high or too low (0) for the circuit to be working, the alarm sounds. A delay should be included between lines 60 and 70 to give time for the heater to have some effect, but the length of time is your own choice.

PROJECT 15

Single-Key Operations Keyboard

The T/S 1000 and T/S 1500 require a number of two-key operations. This project (actually four projects) provides a keyboard that allows single-key operation of four two-key operations—Graphics (G), Function (F), Edit, and Delete. Normally these operations require that the SHIFT key be held down while keys 1, 9, 0, or ENTER are pressed. All of these functions can now be put on individual keys. This saves time in programming and correcting mistakes. It can even be used to move the cursor via SHIFT 5, 6, 7, or 8. Other combinations are possible as long as the KEYBOARD DATA line (KBD0-KBD4) and ADDRESS line (A8-A15) are known for that key.

Materials Required

<i>Quantity</i>	<i>Material</i>
2	Opto-isolator, TIL111
2	Resistor, 1.2 k Ω , 1/4 W
1	Switch, momentary contact
1	Power supply, +5 and +12 V dc
	Project case, assorted hardware and parts as required

Construction

Figure 15-1 is the schematic diagram. The TIL111 IC used in the construction of this project can only be damaged by the

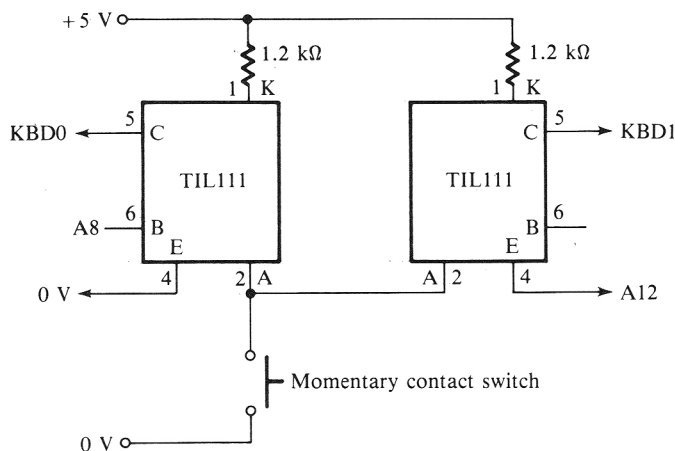


Fig. 15-1

heat of the soldering iron. Reversing any of its pins will only achieve one thing: the device will not work!

Use a 14-pin IC socket to take both ICs, and solder all the wires to the correct pins on the IC socket. Then insert the TIL111s after all the wiring has been checked. The circuit requires a supply of +5 to +9 V, which can be obtained from the computer's own +5-V supply on the edge connector (pin 1B) or the +9-V supply next to it (pin 2B). The connections are numbered from the end nearest the slot in the expansion port; B stands for the fact that they are the underneath (bottom) connections (see table). This supply, however obtained, must be kept away from the rest of the keyboard wiring because it can crash the computer. If the +9-V supply is used, you may even blow up the computer's power pack!

The supply, therefore, should go directly to the board

<i>Operation</i>	<i>ADDRESS (A)</i>	<i>KEYBOARD DATA (KBD)</i>
Graphics	12	1
Function	14	0
Edit	11	0
Delete	12	0

that you are mounting the unit on, and be soldered directly to the junction of the two resistors. Also, take care that there is no way the power supply can touch the 0-V line. The 0-V connection is taken from pins 4B and 5B, which are just to the left of the slot if viewed from the back of the computer. The switch should be a proper push-to-make type and, as it is used regularly, it is best to get a good one. If you are attaching it to the original keyboard, the key may be mounted either next to or on top of the box containing the extra circuitry.

How It Works

The TIL111 is an optical-isolator and contains an LED and a photo-transistor mounted in an IC-type package. This package only has six pins, three on each side, but they are the same width as an ordinary IC. The LED has no physical connection with the photo-transistor; they are just mounted close to each other so that the light from the LED strikes the photo-transistor.

When the LEDs are turned on by operating the switch, light reaches the photo-transistor. This causes the photo-transistor's resistance to drop to 500 Ω , enough to operate the computer's keyboard, connecting both sides of the switch that would normally be pressed by you. As both LEDs are operated by the switch, the computer thinks that two keys have been pressed at the same time. In the circuit diagram given, this is the SHIFT and the 9 key. When the keyswitch is operated, the cursor changes to an inverse G showing that you are now in the GRAPHICs mode.

When the key is released, however, the LEDs turn off and the photo-transistor's resistance increases to more than a million ohms. This makes the computer think that the key has been released and it starts scanning for another key press. The computer will only accept one key combination at a time, so pressing any combination other than SHIFT and another key will be ignored. So don't hold down this key or any of the other forty-odd keys, or it will ignore both.

If the key is wired up to the connections shown in the

“shift-lock” project (Project 13), then the existing keyboard will be affected and can be used as well.

The photo-transistor is very much like a diode, meaning it will pass current only one way, from collector (C) to emitter (E). So the emitter must be connected to an ADDRESS (An), and the collector to the appropriate KEYBOARD DATA line (KBDn). The LED must be connected to the power supply correctly, that is, the cathode to the keyswitch, and the anode to the resistors. The cathode is identified as the point of the arrow with the bar across it. The table shown gives the ADDRESS lines and the KEYBOARD DATA lines for each key attached to OPT 2; OPT 1 is wired up the same way, as it imitates the SHIFT key. LEDs don't like to work with each other and if you put two in parallel, only one will light due to the fact that the first one to light will bypass the other by drawing all the available current; so two resistors are used to overcome the problem.

PROJECT 16

The Moveable "Occupant"

The idea of this project is to do what no other electronic device can do: simulate a human being at home. When you are away from home for a day or more the computer can make your home appear to be occupied because the lights will go on and off at irregular times. In addition, a radio will be heard at intervals, creating the illusion that someone is at home. The unit may cost more than any other project in this book, but it is extremely simple to make and will more than repay the cost by giving you peace of mind.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	OUTPUT port
1	Lamp outlet, 120 V
1	Solid-state relay, 2.5 A, 120 V, with opto-isolator and surge protector
2	Transistor, 2N697
2	Resistor, 2.7 k Ω , 1/4 W
1	Battery connector for 9-V battery
1	Power supply, +5 V dc
	Mounting board, wire, and assorted parts as required

Construction

We are dealing with two voltages here, +5 V dc and 120 V ac. Great care must be taken to construct this project cor-

rectly. *Do not plug in the device until all testing has been completed.* See Figure 16-1 for a diagram.

The solid-state relay comes totally enclosed in a plastic housing with four pins marked 1, 2, 3, and 4. Pins 1 and 2 are the 120-V line terminals, which should be inserted in the hot side of the line (L) to the 120-V lamp outlet. The best place to mount it is near the lamp outlet itself, as this means that the 120-V wiring is kept as far away from the computer as possible. The neutral (N) is connected directly from the line plug to the lamp outlet. Any lamp in the room can then be plugged into the outlet. Make sure that the largest bulb used is 60 watts.

Pins 3 and 4 are the connections to the LED within the unit and must be connected correctly. Pin 4 goes to the collector (C) of the transistor. Pin 3 goes to the +5-V supply. Two wires are connected to pins 3 and 4, then led away from the lamp outlet to a separate board containing the port and the two 2N697 transistors. These wires should be kept well away from the 120-V wiring and should *not* be run back to the computer alongside the line cord.

The lamp outlet should be screwed down to a board to stabilize it. It can then be moved to the room selected and left on the floor or table near the lamp to be plugged in. Remember to switch the lamp on when using the lamp outlet.

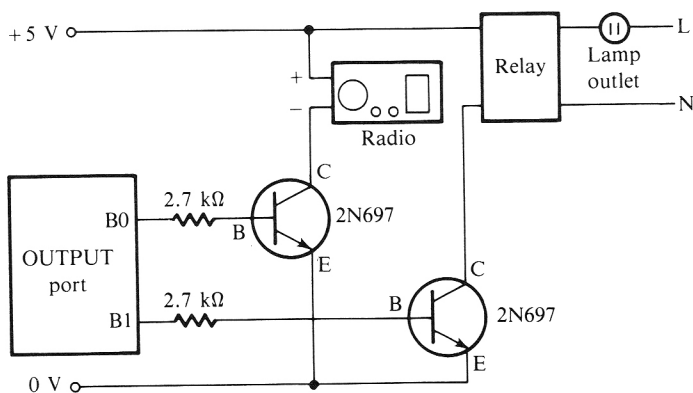


Fig. 16-1

The radio can be in either the same or a different room as the lamp. The cables connecting the radio should be kept short, so I suggest that you keep it in the same room as the computer. The +5 V should *not* be taken from the computer's own +5-V supply, as surges created by switching the radio on and off may lead to crashing the computer's program.

The transistors used are 2N697s and these have the collector connected to its metal casing. Therefore, the cases of the two transistors must be kept apart. The usual warnings about soldering transistors apply, as does making sure they are wired correctly. The radio used should be of the portable type, run by a 9-V battery. Larger types might draw too much current from the +5-V supply, and the maximum the transistors can take is only 500 mA before any damage is done.

How It Works

The program times a 24-hour period and can switch on either the radio or the lamp, depending on the time of day. This is to make sure that the "occupant" does not disturb the neighbors. The port data either turns on the radio by setting B0 to binary 1, or turns on the lamp by setting B1 to binary 1. This is done by **POKE**ing the port with either 1 or 2. If required, the nighttime occurrences could **POKE** the port with 3, thus turning on the lamp and the radio. If your port requires the use of a machine code routine to get information into it, then substitute the appropriate **POKE**, followed by the call to the routine, via the **USR** command.

The transistors reduce their emitter-to-collector (E-to-C) resistance to about 0 Ω when the +5 V from the port is applied to its base (B) (binary 1). The 2.7-k Ω resistor stops the damage to the transistor and the port bit by limiting the amount of current flowing into the base. Therefore, when the port turns that bit to binary 1, the collector is virtually at 0 V. This turns on the radio which you will already have tuned into a station and adjusted the volume control. On the components list is a battery connector, which should be used to connect the radio. This should be wired in reverse to normal, that is, black to the positive (+) supply and red to the neg-

ative (–) collector connection. This is because it replaces the battery used for the radio.

The relay acts as the line switch and is operated by the collector/emitter resistance being at 0 V. Within the relay there is now enough voltage across an internal LED to light it. The LED operates a photo-transistor, which in turn operates an ac switching device, called a *triac*. A triac can be operated at any time during the ac cycle, meaning that it can switch on at any voltage. It acts just like a light switch. The instantaneous voltage at switch ON can vary from –120 V to +120 V. If the instantaneous voltage is very high, then a noise “spike” is created which bounces through all the electrical wiring in the house. Since the computer is also connected to the line through the transformer, it also receives this high voltage spike. This spike will upset the computer by creating a sudden surge that will corrupt the RAM and crash the program. So switching things on and off can easily crash the computer. When we need to control things on the line by computer, we need a way of stopping these spikes from being created.

Such a device is called a *zero voltage switch* and it is contained in the solid-state relay. This will turn on the triac only when there is no voltage (no current flowing), that is, every time the voltage changes from positive to negative and back to positive again. As there is no voltage or current flowing, no spike can be produced and the voltage rises from 0 V to the maximum voltage, as normal. This means that when we turn on the internal LED, we may have to wait up to 1/100th of a second (10 ms) before the relay’s triac is turned on. But because this prevents the program from crashing, I think it is worthwhile. It will also deal with the spikes that are created when controlling electric motors.

Because of the zero switching facility being built in, the device can quite safely be operated by the computer. There is no physical contact between the 120-V line and the port circuit. The relay is rated at 2.5 A and that means that it can safely cope with inductive loads (motors, coils, and the like) and resistive loads (lamps, heaters, and the like) up to 750 W.

The Program

```

10  REM SET UP PORT IF NECESSARY
20  PRINT "INPUT HOUR 1-24"
30  INPUT N
40  IF N>24 OR N<1 THEN GOTO 30
50  PRINT " WORKING "
60  FOR Y=N TO 24
70  IF Y>8 AND Y<16 AND RND*8>.5 THEN
    POKE PORT, 1
80  IF Y>19 AND Y<23 AND RND*8>.5 THEN
    POKE PORT, 2
100 GOSUB 200
110 POKE PORT, 0
120 NEXT Y
130 LET N=1
140 GOTO 60
200 FOR H=1 TO 60
210 PAUSE 3000
220 NEXT H
230 RETURN

```

The timing is being done in this program by the subroutine 200, which **PAUSEs** for $3000 \times 1/50$ th second, which is approximately 1 minute. This is done 60 times, so that call to **GOSUB 200** provides a delay of 1 hour before any device is switched off by the **POKE** in line 110. All timing is done to a 24-hour clock, so 1 o'clock in the afternoon is 1300 hours, and so on.

PROJECT 17

A Computerized Voltmeter

The circuit given here enables you to use the A/D converter project to measure voltages. These voltages can be anywhere from +0.5 to +3.5 V. This range is very small, but is larger than most A/D converters. It can only be used to measure positive voltages with respect to the 0-V line. It can, however, be extended to cover most voltage ranges found on the normal voltmeter by adding suitable external resistors, called *multipliers*, to drop the extra voltage.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Transistor, NPN, BC108
1	A/D converter
1	Resistor, 82 k Ω , 1/4 W
1	Resistor, 1 k Ω , 1/4 W
	Project case and assorted hardware as required

Construction

The transistor circuit (Figure 17-1) can be built as a separate unit, or on the same board as the A/D converter. The wiring to the connections of the A/D converter from the transistor should not be greater than a few feet, as the reading depends on the resistance of the transistor across terminals A and B. A diode might have to be included for protection of the tran-

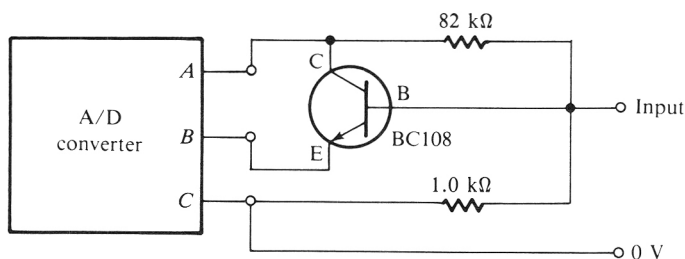


Fig. 17-1

sistor if the voltages could go negative at any time. The cathode of the diode is connected to the base of the transistor, and the anode is connected to the voltage to be measured (or to the multiplier, if one is used). The cathode of a diode is the tip of the arrow on the diode symbol, and is marked with a band on the actual body of the diode.

This instrument is not as accurate as a multimeter (also known as a multitester), which has an internal resistance of 20 kΩ per volt or 60 kΩ on a 3-V scale. The voltmeter described has an internal resistance of only 1 kΩ per 3 V, therefore, it puts resistance in any circuit it measures, causing the voltage to drop. This is how you can get false readings. The voltmeter will, however, be more accurate than the multimeter insofar as a small change in voltage will make a greater change in the number returned to the computer than will be indicated on the multimeter.

The value of the two resistors may be changed to give the voltmeter a higher internal resistance, but the ratio of 82/1.0 must be maintained.

How It Works

A transistor is used as a variable resistor, as the resistance between its emitter (E) and the collector (C) varies with the voltage on its base (B). The silicon transistor (the type used in this project) shows a very high resistance when the base voltage is below 0.6 V and has virtually no effect on the 100-

k Ω resistor across terminals A and B of the A/D converter. This 100-k Ω resistor controls the timing and in turn, controls the number returned to the computer. If the voltage rises above 0.6 V, the transistor's resistance across the 100-k Ω resistor starts to come down at the same rate that the voltage rises (Figure 17-2). As the 100-k Ω resistor is bypassed more and more by the lower resistance of the transistor, more voltage reaches the timing capacitor, causing it to charge more quickly. Therefore, the higher the voltage, the shorter the duration of the pulse from the A/D converter. This means that the higher the voltage, the lower the number returned to the BASIC program from the machine code collecting routine.

As any voltage below 0.6 V has virtually no effect on the timing, I decided to set the lower limit to be measured at a constant 0.6 V. This is done to keep the transistor permanently in the varying resistance range; it will then react to any small increase in voltage above 0.6 V. This is done by having two resistors fixed to the base of the transistor. The other ends of the resistors are connected to the +5-V supply and the 0-V line of the A/D converter. This is called a voltage divider because it divides the voltage according to the ratio of the resistors. The base of the transistor has virtually no effect on this calculation and can be ignored. As the 82-k Ω resistor is the higher value of the two, it drops the most volt-

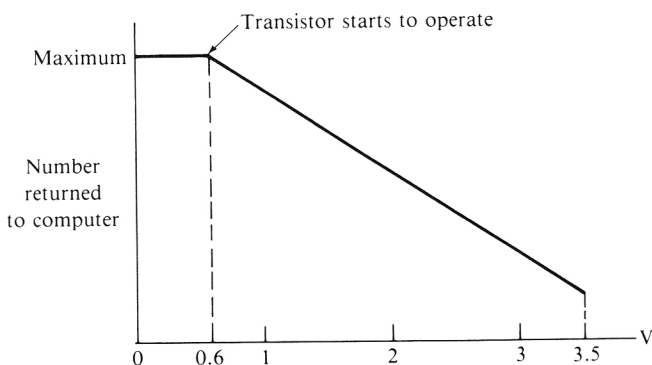


Fig. 17-2

age across itself. Therefore, between the base and the 0-V line, the voltage is slightly more than 0.6 V.

The unit has to be calibrated by setting up a voltage from an external power supply and by using a multimeter that has been set to the voltage range. The voltmeter should be used to measure the voltage at the base of the transistor as the numbers from the machine code routine are noted. The power supply should be increased in steps of 0.5 V and the readings should be stored in the computer for later reference. The 0.5-V steps can be calculated from the readings obtained since the difference between them is in linear form; in other words, the number varies in direct proportion to voltage applied.

To extend the ranges, a suitable resistor must be put in series with the input voltage. The resistance is calculated from the following voltage ratio:

$$\frac{\text{Maximum voltage}}{3.5 \text{ V}} = \frac{X \Omega}{1.0 \text{ k}\Omega}$$

$$\text{Therefore, } X = \frac{\text{Maximum voltage} \times 1.0}{3.5}$$

The answer is in kilohms.

The maximum range is limited because the transistor is almost fully turned on and a small change in voltage makes almost no difference in the number collected by the computer. A range from 5545 to 960, with an accuracy of ± 1 on the count is feasible. This means considerable sophistication of the measurement of voltages within the 0.6- to 3.5-V range.

PROJECT 18

Unbeatable Burglar Alarm

This is yet another project that uses the A/D converter project. In this one the A/D is used to measure the resistance of a series of resistors that are connected to switches attached to doors, windows, and the like. Each switch is closed by the opening of a door or window which sets off the alarm. If the alarm wire is cut, the alarm goes off. If the switch is short-circuited by wiring across the switch with a piece of wire, the alarm goes off. If the resistor is spotted and the resistor is substituted by another, the alarm goes off. As long as the computer is on, the system is hard to beat. It can also tell you exactly where the alarm has been set off from the literally hundreds of switches that could be attached to this one burglar alarm. It only requires one set of wires, too!

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Resistor, 10 k Ω , 1/4 W
1	Resistor, 15 k Ω , 1/4 W
1	Resistor, 18 k Ω , 1/4 W
1	Resistor, 120 k Ω , 1/4 W
1	A/D converter
4	Microswitch
	Project case and assorted hardware as required

Construction

The A and B terminals of the A/D converter in Figure 18-1 are connected to a chain of resistors. Each resistor is then shunted by a microswitch (see Figure 18-2). The switches must be set to close when an object is moved, or a door or window

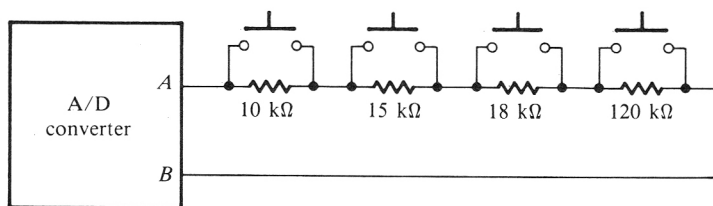


Fig. 18-1

is opened; how it is used is left up to you. A few suggestions for alternatives to the microswitches are:

1. Tilt switches that operate when they are tipped more than 30 degrees.
2. Reed switches operated by magnets in the door or window.
3. Pressure mats under carpets.

The resistor should be soldered across the switch with a connection that is as short as possible, so the switch cannot be cut away from the resistor while leaving the resistor in the circuit. Attach the wires so there is only one wire going into and one wire coming away from the switch; it makes it easier to identify later if you want to add to or change the circuit in any way.

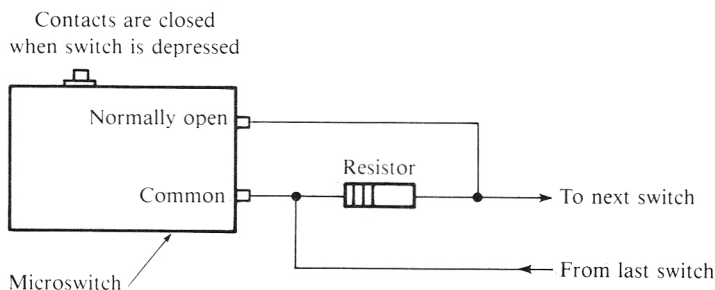


Fig. 18-2

The wire can be any length (as long as it doesn't run into miles), as this can be compensated for within the program. Each switch is identified by having a resistor of a unique value. The joints between the wires and the microswitch should either be soldered or tightly wire-wrapped; otherwise, they might set off the alarm by changing its resistance. The program should be adjusted by setting off the alarms after they are all installed since the program will be different with the addition (or deletion) of each switch.

How It Works

The A/D project measures the resistance across the terminals A and B by seeing how much the 100-k Ω resistor within the unit is bypassed. Therefore, the first reading must be taken to see what the normal resistance is. This should be checked by the program, and only if it is different from normal should any checks be made on what has caused the alarm to operate. This saves time since the checks could run into minutes if quite a few switches are used. The normal response time of the A/D converter and machine code program is less than 50 ms, leaving little time to modify the circuit or bypass it. The total resistance of the circuit is measured continuously, but when one of the switches is operated, even momentarily, the total resistance changes because one of the resistors has been removed from the circuit. The total resistance of the chain can be calculated by adding up all the values of the individual resistors. The total resistance should not exceed 10 M Ω nor be less than 11 k Ω . The resistors can be of any tolerance since the individual resistances are unlikely to be exactly the same. If you run out of preferred values, that is, the ones that are normally available, you can join two together to get additional values.

The resistance can be checked to see which switch has been operated by taking its value away from the normal value (when no switch is operated). The result should be compared against the value recorded when the system was set up. If it matches, within ± 5 , the alarm location should be printed on the screen and an alarm should be sounded (using the OUT-

PUT port to operate a tone generator, bell, or something similar). The number of locations covered by a 2K computer will obviously be limited; a 16K model would be required for a large installation. The program could also include variations designed to your own specifications; for example, an occupied bedroom could be protected. It could also incorporate a timer, which would allow further sophistication of the system. The choice is left up to you to write your own program to incorporate ideas that will be unique to your house, business, and the like. Even buying a T/S 1000 or T/S 1500 and building or buying a port, especially for this project, costs less than some simple alarm systems. It is also easily adapted, and can be reprogrammed at any time.

PROJECT 19

Battery Power Supply

This useful addition to a T/S 1000 or T/S 1500 system will not only eliminate those annoying “whiteouts” due to fluctuations in the 120-V line, but also provide a completely portable system independent of any power supply. The unit supplies +12 V from two batteries, with enough power to supply a TV and computer for up to 3 h.

The amount of power required for the system is calculated in watts and is achieved by multiplying the maximum amount of current required in amperes by the voltage supplied. The computer might take as much as 1 A from a 12-V supply. Thus it would need a 12-W supply (12×1). A typical small screen dc-powered black-and-white TV takes only 7.5 W from a 12-V supply. That means that the total computing system takes only 19.5 W. I have omitted the cassette recorder because it is usually a portable one with its own built-in battery supply.

Batteries are rated by the number of amperes (A) that they can supply over a given period, normally quoted in ampere hours, that is, 1 A for 1 h. If a battery is rated at 6 A·h, theoretically it can supply a maximum of 6 A for 1 h or 1 A for 6 h (from a practical standpoint, this isn't the case). A car battery could be used to provide enough ampere hours for the system, but this would be extremely heavy and costly. Also, car batteries can be messy and dangerous since they are filled with sulfuric acid. One solution would be to use a portable battery pack such as those used

for TV sets and video cameras. A less expensive alternative would be to use two rechargeable 6-V lantern batteries.

As we need 12 V, the two batteries must be connected in series, that is, the plus terminal of one is connected to the minus terminal of the other. The battery set is normally left attached to the computer via a 3.5-mm plug. When the computer is switched off, the batteries can be recharged via the 20- Ω resistor. The resistor is necessary as the batteries must be charged at a slow rate. The diode should always be included and it should have as high a rating as possible so that it will drop little voltage across its own internal resistance.

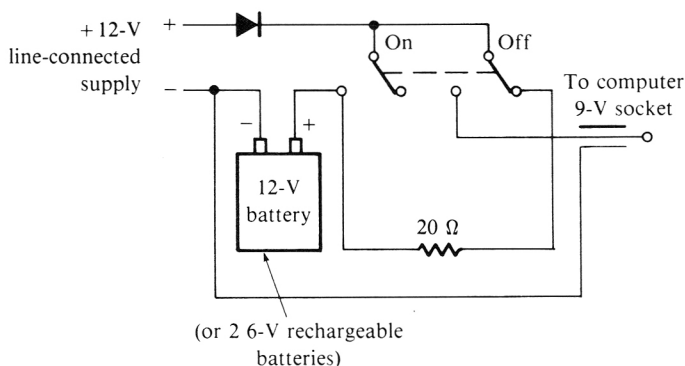
If the line-connected power supply can provide more than 1.6 A, the batteries can be recharged while using the computer with its power supply unit. The batteries should be left to charge from the line power supply, with the computer system switched off, for 10 to 12 h. The diode ensures that the batteries cannot discharge through the line supply when the power supply is switched off because it only allows the current to flow in the direction of the arrow in the diode symbol (positive to negative), and stops the flow of current from the positive of the battery set back to the negative.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	Battery, 12 V, nickel-cadmium rechargeable, 6 A·h (or two 6-V batteries)
1	Power diode, 5 A
1	Switch, double-pole double-throw
1	Resistor, 20 Ω , 5 W
1	Jack plug, 3.5 mm
1	Power supply, +12 V, 1 A
	Project case and assorted hardware as required

Construction

Figure 19-1 is the schematic diagram. The 3.5-mm plug must be wired correctly with the “knob” connected to the positive side of the supply (+). The switch should be rated at least

**Fig. 19-1**

2 A and is a double-pole, double-throw switch. The switch may also be used as a reset switch to save wear on the 3.5-mm jack socket on the computer. It is best not to use a slide switch as they tend to be unreliable after short periods of heavy current, causing a sudden power loss.

The rest of the construction is not critical; the layout can be of your own choice, except to wire the diode and the battery correctly. The positive end of the diode is usually the end marked with a band.

PROJECT 20

Power Supply Filter

A power supply filter is an essential component of many electronic devices from the humblest microcomputer to the mightiest mainframe. The T/S 1000 and T/S 1500 tend to suffer from "whiteouts" caused by interference from the power supply, which can corrupt the data held in the RAM.

This interference comes in many forms: (1) motor circuits turning on or off cause surges due to their large inductance, (2) radio interference is picked up from overhead wires or even the house wiring, and (3) sudden surges caused by large changes in current due to the operation of heaters and television sets.

This filter cuts out interference in the line by allowing only low frequencies through, such as the 60-Hz ac line voltage. Any high frequencies have difficulty getting through for two reasons:

1. At the higher frequencies, the filter presents a high impedance in series with the computer, thus blocking out these high frequencies.
2. At the higher frequencies, the filter capacitors connected across the 120-V line present a low impedance, so the capacitors tend to short out higher frequencies.

The trouble caused by high frequency noise "spikes" is mostly eliminated by the filter. It will not, however, cope with

large fluctuations in line voltage for more than a few milliseconds.

Materials Required

Quantity	Material
3	Capacitor, 0.1 μF , 600 V ac
4 ft	Magnet wire, No. 18
	Project case, terminal blocks, line cord, male plug, female line socket, strain relief, assorted hardware as required

Construction

The schematic diagram is shown in Figure 20-1. The 120-V ac line can be lethal and needs to be handled with care and consideration. *Do not ever work on any powered equipment without first removing the plug from the wall socket.*

The metal case housing the filter *must* be connected to ground; it will protect you by blowing the fuse if the line touches the case. The filter should be placed as near to the T/S 1000 or T/S 1500 power pack as possible because the line cord can also act as a radio antenna, picking up inter-

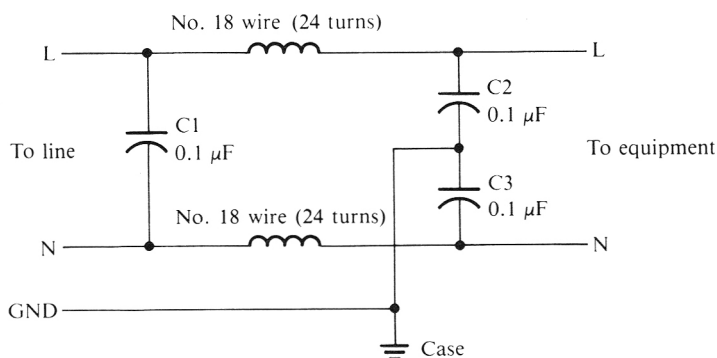


Fig. 20-1

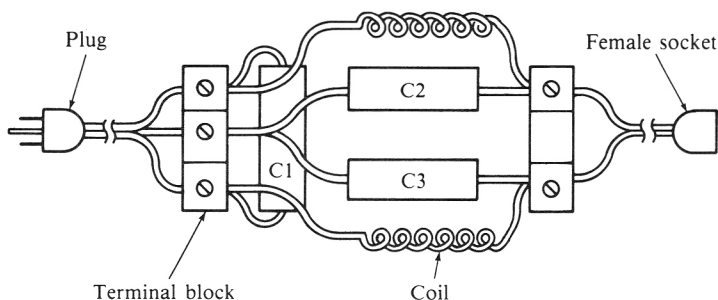


Fig. 20-2

ference. The case must be made out of metal to shield the two coils and prevent them from acting like a radio receiver, picking up more interference.

The 120-V ac line is connected to the unit inside the case by plastic terminal blocks. The blocks can be secured to the metal chassis with screws. The capacitors should be arranged as shown in Figure 20-2, as they form a support for the coils.

The coils are wound from 4 feet of No. 18 enamelled magnet wire on a 1/4-inch diameter wooden rod. The 24 turns are wound as close together as possible along the rod. After the coil is removed from the rod, allow the turns to naturally spring apart so that none of the turns touch each other. This should leave you with a coil of wire about 2 inches long. After scraping the enamel off to bare the copper on each end, solder the coils directly to the capacitors. Make sure that the coils are at least 1/4 inch away from any part of the metal case. All of the leads in this unit should also be as far away from the metal case as possible.

A cadmium-plated chassis or project box is excellent for housing this filter. An aluminum box of at least $4\frac{1}{2} \times 3 \times 1$ inch would also do, but aluminum cannot be joined with ordinary solder, so you will have to solder all the case connections to solder terminals and then screw the terminal board to the box.

If you have an ohmmeter, check that there are no shorts to the case with the cover on; if you don't, make sure you check it visually.

PROJECT 21

A Logic Probe

The logic probe tests to see if the condition on a circuit is binary 1 or binary 0, lighting either a red or yellow light-emitting diode, thereby testing the T/S 1000 and T/S 1500.

If the condition being tested is a series of pulses, both LEDs will flash. This can be used to test any logic circuit, including that of the computer without affecting its operation. It can also be operated using the power supply from the computer.

Materials Required

<i>Quantity</i>	<i>Material</i>
1	LED, yellow
1	LED, red
1	Resistor, 1.2 k Ω , 1/4 W
1	Resistor, 220 Ω , 1/4 W
1	IC 7400
1	IC socket, 14-pin
1	Power supply, +5 V dc
	Small length of No. 18 wire, ball-point or felt-tip pen case, hook-up wire, and assorted parts as required.

Construction

The housing of this project is a felt-tip or ball-point pen case, on top of which is mounted the IC socket containing the 7400 IC. See Figures 21-1 and 21-2. The wires connecting the IC

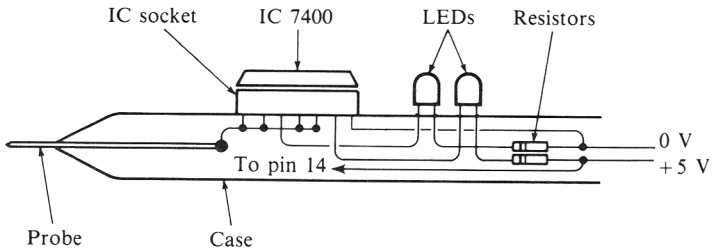


Fig. 21-1

socket should first be soldered to the components inside the case. Then, make a hole in the side of the case for the connection of each pin, and push a wire through each hole and solder it to the IC socket. The LEDs should be mounted by the same method through holes in the case, this time, however, the wires should be soldered onto the LEDs first. Then draw out the wires at the end of the pen and solder them to the resistor before the IC socket is wired. The probe should be made out of a short piece of bare No. 18 AWG wire which is soldered to another thinner hook-up wire, which is in turn soldered to pin 1 of the IC socket. Glue the No. 18 wire to the end of the case. If the wire you have has enamel or another insulating coating, scrape off the insulation at both ends of the wire probe so that a good contact with the circuit and the soldered wire can be made. There are only eight pins to solder on the IC socket.

The current drawn by the probe from the power supply is only 20 mA, so it can be attached to the computer's +5-V output on the expansion port, if necessary. The +5-V supply is obtained from pin 1B on the computer's edge connector (on the bottom of the edge connector). On the T/S 1000 it is the pin closest to the TV socket; on the T/S 1500 it is nearest the MIC socket.

The IC should not be plugged into the socket before all of the connections have been soldered and checked. This saves possible damage to the IC.

When the power supply is connected and nothing has been attached to the tip of the probe, the red LED should light.

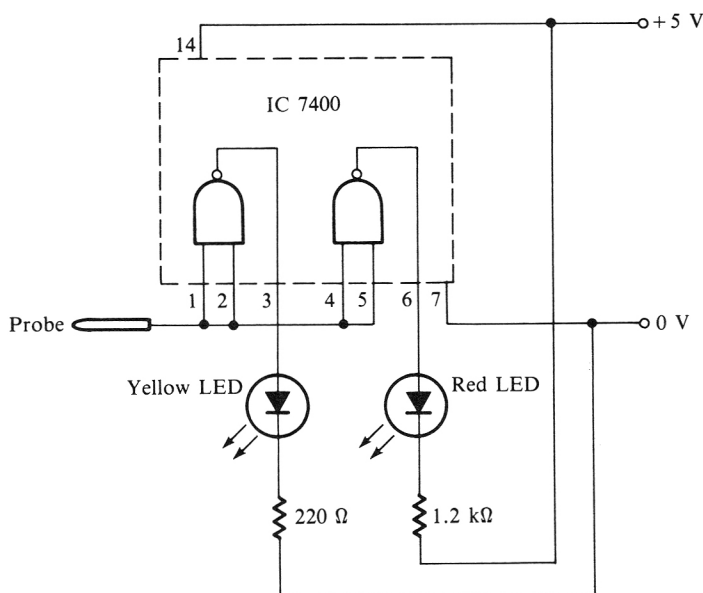


Fig. 21-2

The power supply, if not connected to the computer's power supply via the 0-V line, *must* be attached to the 0-V line on pins 4/5B on the edge connector. This could be done by attaching a test clip to the TV lead's outer metal conductor, which is also at 0 V, if you don't want to solder a connection. An injector-type test prod can be used in the small exposed area. When testing circuits that are not connected to the computer, attach the 0-V line of the power supply to the 0-V line of the circuit to be tested.

How It Works

The probe's tip is connected to a point on the circuit that is at one of two levels, binary 1 or binary 0. This is +5 V and 0 V on the 7400 IC. All the projects in this book use +5 V. There are other types of ICs that use different methods and voltages to communicate binary 1s and 0s, but a class of

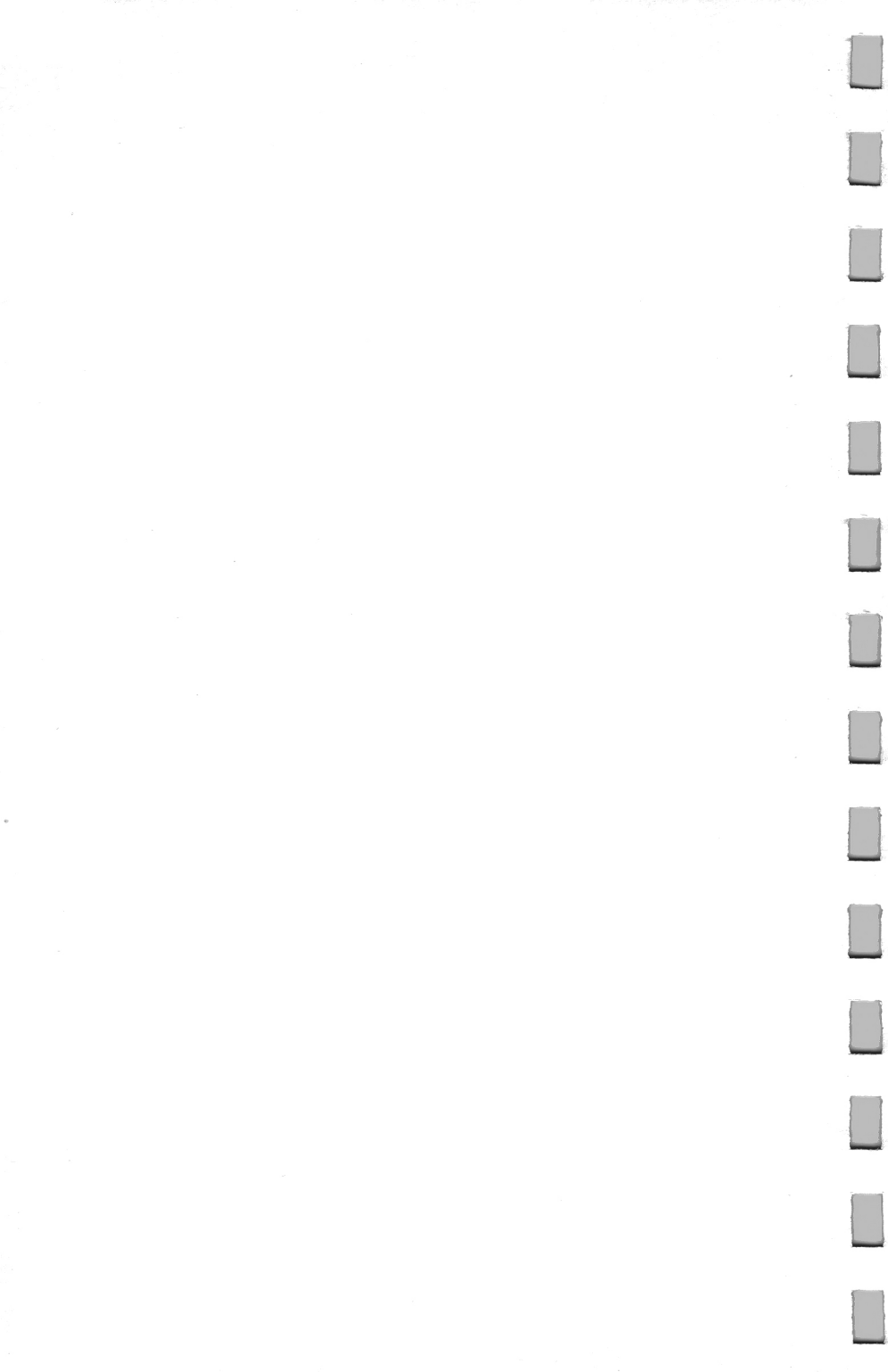
ICs known as transistor-transistor-logic (TTL) is the most common. These others cannot be tested because the voltages might be greater than the input voltage allowed for these 7400 ICs and they would damage the tester.

A binary input to both pins 1 and 2 causes the NAND gate to react like an inverter, meaning if a binary 1 is input, the output is binary 0, and vice versa. When the input is binary 1, the output is 0 V (binary 0) and the yellow LED will not light because it doesn't have a sufficient difference across it to light, both ends being at 0 V. The red LED, however, has +5 V at the end of its resistor and will light because its gate inputs also cause it to change the output to 0 V. When the input is changed to binary 0, the yellow LED will glow and the red LED will go out. Therefore, the red LED gives an indication of a binary 1 on the input, and a yellow LED gives an indication of binary 0 input. When the circuit to be measured is constantly changing between binary 1 and binary 0, or pulsing, both LEDs will start to flash at the rate which the pulse changes. If the pulsing rate is too high, all that will be seen on the LEDs will be a faint flicker.

There are many uses of the logic probe. These include finding out whether the correct ADDRESS is operating the correct $\overline{CS}$ pin, as in the "Wheel of Fortune" project (Project 10), checking that none of the ADDRESS lines are short-circuited; and checking the data sent to a port on a particular bit, to name but a few.

One thing to remember is that the logic probe will indicate no connection by binary 1. This is because the 7400 IC keeps all its inputs at binary 1 by means of resistors within the chip connecting the +5-V supply and the input.

The resistors used, reflect the various resistances involved inside the output of a gate; the 220- Ω resistor is smaller than the 1.2-k Ω resistor because of the resistance through which it must travel within the chip from the +5-V line. This resistance can be up to 10 k Ω , so a small resistor is needed to keep the current down to a reasonable level. The resistance to the 0-V line, when the output is binary 0, is only that of a transistor's emitter/collector junction. This is very low when turned on, so a larger resistor must be used.



Catalog

If you are interested in a list of fine Paperback books, covering a wide range of subjects and interests, send your name and address, requesting your free catalog, to:

McGraw-Hill Paperbacks
1221 Avenue of Americas
New York, N. Y. 10020

BEEPERS

21 Electronic Projects for the

TIMEX/SINCLAIR 1000 and 1500

Gordon Rockmaker and Stephen Adams

This book contains twenty-one easy-to-build electronic projects for the Timex/Sinclair 1000 and 1500. With just a few simple tools and skills, enumerated in the book, readers can construct accessories for the computer (such as a numeric keypad and a shift-lock device) as well as expand its interface with appliances in the home and office.

Each of the projects is explained with clearly written step-by-step instructions, including materials lists, schematic diagrams, and suggestions for obtaining parts. Using this book, even the novice in electronics will be able to harness the power of the T/S 1000 and 1500 to monitor security systems, control temperatures, measure electrical quantities, and learn digital electronics.

Gordon Rockmaker is Editor-in-Chief of Electricity, Electronics, and Computer Technology in the Gregg Division of the McGraw-Hill Book Company.

Stephen Adams has written numerous articles on the Sinclair ZX80 and ZX81 computers and has been an active member of the National ZX80 and ZX81 Users' Club in his native England.



ISBN 0-07-053358-X